

Tiedekunta/Osasto — Fakultet/Sektion — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen		Matematiikan ja tilastotieteen laitos	
Tekijä — Författare — Author Topias Rusanen			
Työn nimi — Arbetets titel — Title Legopalikoita ja 4D-röntgentomografiaa			
Oppiaine — Läroämne — Subject Matematiikka			
Työn laji — Arbetets art — Level Pro gradu -tutkielma		Aika — Datum — Month and year Lokakuu 2018	Sivumäärä — Sidoantal — Number of pages 43 s.
Tiivistelmä — Referat — Abstract <p>Röntgentomografiassa on tarkoitus hankkia tietoa jonkin tutkittavan kappaleen sisärakenteesta rikkomatta itse kappaletta. Tomografialaitteet ottavat kappaleesta paljon röntgenkuvia eri suunnilta. Kuvien perusteella on mahdollista laskea esimerkiksi poikkileikkauskuva kappaleesta. Jos röntgen-dataa on tarpeeksi, pystytään kuvien avulla luomaan kappaleesta kolmiulotteinen malli. Tätä mallia voidaan tarkastella mistä kuvakulmasta tahansa. Poikkileikkauskuvia ja kolmiulotteisia malleja kutsutaan <i>rekonstruktioiksi</i>, sillä sisärakenne "rakennetaan uudelleen" röntgenkuvista.</p> <p>4D-tomografiassa mukaan tulee myös aikaulottuvuus. Röntgendataa kerätään usealta ajanhetkeltä ja rekonstruktioiksi muodostuu animaatio, joka kuvaa kappaleen rakenteen muutosta ajan suhteen. Myös tätä animaatiota pystytään tarkastelemaan mistä kuvakulmasta tahansa minä ajanhetkenä tahansa.</p> <p>Tämän opinnäytetyön tarkoitus on soveltaa kahta eri algoritmia rekonstruktioiden laskemiseen jokaiselle ajanhetkelle erikseen. Liikkuvasta LEGO-robotista on tarkoitus saada aikaiseksi 4D-rekonstruktio. Yhtenä ongelmana on, muodostavatko yksittäiset rekonstruktiot järkevän kokonaisuuden, kun ne asetetaan peräkkäin? Entä millä tavalla eri algoritmeilla lasketut rekonstruktiot eroavat toisistaan?</p>			
Avainsanat — Nyckelord — Keywords 4D-Röntgentomografia, SIRT, CGLS, Conjugate gradients, LEGO			
Säilytyspaikka — Förvaringsställe — Where deposited Kumpulan tiedekirjasto			
Muita tietoja — Övriga uppgifter — Additional information			

LEGO-palikoita ja 4D-röntgentomografiaa

Topias Rusanen

16. lokakuuta 2018

Sisältö

1	Johdanto	2
1.1	Merkinnöistä	4
2	Aineisto ja ohjelmistot	5
2.1	Mittausjärjestelyt ja aineisto	5
2.2	Ohjelmistot ja rauta	6
3	Menetelmät	8
3.1	Matemaattinen perusta	8
3.2	Lambert-Beerin attenuaatiomalli	8
3.3	Viivaintegraalit ja niiden mittaaminen	10
3.4	Diskretointi ja lineaarinen malli	12
3.5	Numeeristen menetelmien ongelmat	13
3.6	Kaczmarzin algoritmi	14
3.7	ART ja SIRT	17
3.8	Kvadraattiset rekonstruktio menetelmät	19
3.9	CGLS	22
3.10	CGLS:n soveltaminen normaaliyhtälöihin	29
3.11	Aikaulottuvuuden normalisointi	30
4	Tulokset ja pohdintaa	32

Luku 1

Johdanto

Mitä on 4D-röntgentomografia? Tämä kysymys nousee heti esiin tämän opinnäytetyön otsikkoa lukiessa. Röntgentomografia on tärkeä matematiikan ala, jolla on tärkeitä sovelluksia esimerkiksi lääketieteessä ja tuotekehityksessä.

Röntgentomografiassa on tarkoitus hankkia tietoa jonkin tutkittavan kappaleen sisärakenteesta rikkomatta itse kappaletta. Tomografialaitteet ottavat kappaleesta paljon röntgenkuvia eri suunnilta. Kuvien perusteella on mahdollista laskea esimerkiksi poikkileikkauskuva kappaleesta. Jos röntgendataa on tarpeeksi, pystytään kuvien avulla luomaan kappaleesta kolmiulotteinen malli. Tätä mallia voidaan tarkastella mistä kuvakulmasta tahansa. Poikkileikkauskuvia ja kolmiulotteisia malleja kutsutaan *rekonstruktioiksi*, sillä sisärakenne "rakennetaan uudelleen" röntgenkuvista.

4D-tomografiassa mukaan tulee myös aikaulottuvuus. Röntgendataa kerätään usealta ajanhetkeltä ja rekonstruktioiksi muodostuu animaatio, joka kuvaa kappaleen rakenteen muutosta ajan suhteen. Myös tätä animaatiota pystytään tarkastelemaan mistä kuvakulmasta tahansa minä ajanhetkenä tahansa.

4D-tomografia on melko uusi tutkimusala. Aiheesta ei löydy vielä paljoa kirjallisuutta. Ensimmäiset kaupalliset sovellukset ovat alkaneet löytää tiensä markkinoille, esimerkiksi GE Healthcare hyödyntää 4D-tomografiaa omassa tuotevalikoimassaan.

Nykyisillä tomografialaitteilla voidaan jo kerätä 3D-rekonstruktioon tarvittava data. Miksi sitten näillä laitteilla ei vain oteta röntgenkuvia usealta ajanhetkeltä ja luoda 4D-rekonstruktioita? Ongelmiksi muodostuvat seuraavat kaksi tekijää: kuvausaika ja säteilyn määrä. Perinteiset rekonstruktioalgoritmit tarvitsevat paljon röntgendataa pystyäkseen tuottamaan tarkkoja rekonstruktioita. Tästä johtuen säteilyn määrä on suuri, sillä röntgenkuvia tarvitaan paljon. Lisäksi kuvan ottaminen kestää kauan. Tomografialaitteissa on monesti vain yksi röntgenlähde ja -detektori. Nämä kiertävät kuvattavan kappaleen ympärillä ja keräävät hitaasti tarvittavan röntgendatan. Röntgenlähteitä ja -detektoreja voidaan lisätä, jolloin kuvausaika nopeutuisi, mutta tällöin myös potilaan saama säteily-

annos kasvaa.

Esimerkiksi lääketieteellisissä sovelluksissa kehon muutokset ovat nopeita. Sydämen sykettä kuvattaessa sydän sykkii helposti useita kertoja sekunnissa. Mikäli tätä liikettä haluttaisiin mallintaa tomografialaitteilla, pitäisi laitteiden kuvausajan olla todella nopea. Lisäksi elävää kohdetta kuvattaessa säteilyn määrän tulisi olla mahdollisimman pieni. Näistä rajoituksista johtuen 4D-tomografia ei ole ollut nykylaitteilla mahdollista.

Modernit rekonstruktioalgoritmit pystyvät tuottamaan tarkkoja rekonstruktioita vain murto-osalla käytettävissä olevasta röntgendatasta. Tällaisia menetelmiä on esimerkiksi Helsingin yliopiston Inversio-ongelmien huippututkimusyksikössä kehitetty "*Level set*"-metodi [6]. Ne toimivat myös esimerkiksi tilanteissa, joissa röntgendataa ei ole saatavilla kaikista suunnista, vaan vain osasta. Koska dataa tarvitaan vähän, pienenee sekä laitteiden kuvausaika että potilaan saama säteilyannos.

Tätä opinnäytetyötä varten kerättiin röntgenkuva-aineisto, joka sisältää 24 820 yksittäistä röntgenkuvaa. Aineisto on 4D-tomografiaan soveltuva ja kuvaa LEGO-robotin liikettä 68:n ajanhetken ajan. Työssä käytetään vanhempia, hyviksi todettuja rekonstruktioalgoritmeja, jotka eivät ota aikaulottuvuutta millään lailla huomioon.

Tämän opinnäytetyön tarkoitus on soveltaa kahta eri algoritmia rekonstruktioiden laskemiseen jokaiselle ajanhetkelle erikseen. Liikkuvasta LEGO-robotista on tarkoitus saada aikaiseksi 4D-rekonstruktio. Yhtenä ongelmana on, muodostavatko yksittäiset rekonstruktiot järkevän kokonaisuuden, kun ne asetetaan peräkkäin? Entä millä tavalla eri algoritmeilla lasketut rekonstruktiot eroavat toisistaan?

1.1 Merkinnöistä

- Vektorit esitetään tässä opinnäytetyössä ylänuolen avulla, esim. \vec{a} on vektori, kun taas a on joko skalaari tai funktio.
- Vektorin $\vec{x} \in \mathbb{R}^n$ komponentteja merkitään alaindeksillä ilman vektoriviivaa:

$$\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

- Numeroituja vektoreita merkitään alaindeksillä vektoriviivan kanssa

$$\vec{x}_1, \vec{x}_2, \vec{x}_3, \dots$$

tai sulkeissa olevalla yläindeksillä

$$\vec{x}^{(1)}, \vec{x}^{(2)}, \vec{x}^{(3)}, \dots$$

- Olkoon $\vec{a}, \vec{b} \in \mathbb{R}^n$. Näiden vektorien euklidista sisätuloa merkitään pisteellä:

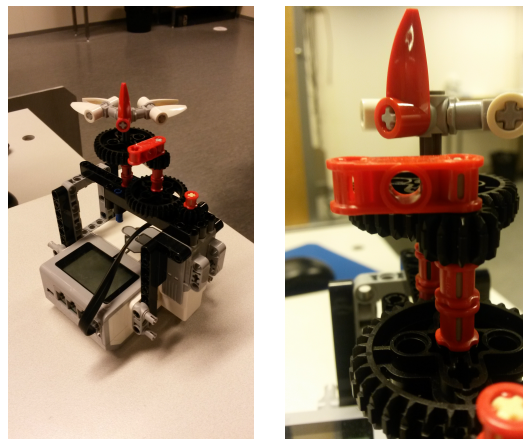
$$\vec{a} \cdot \vec{b} = a_1 b_1 + \dots + a_n b_n$$

Luku 2

Aineisto ja ohjelmistot

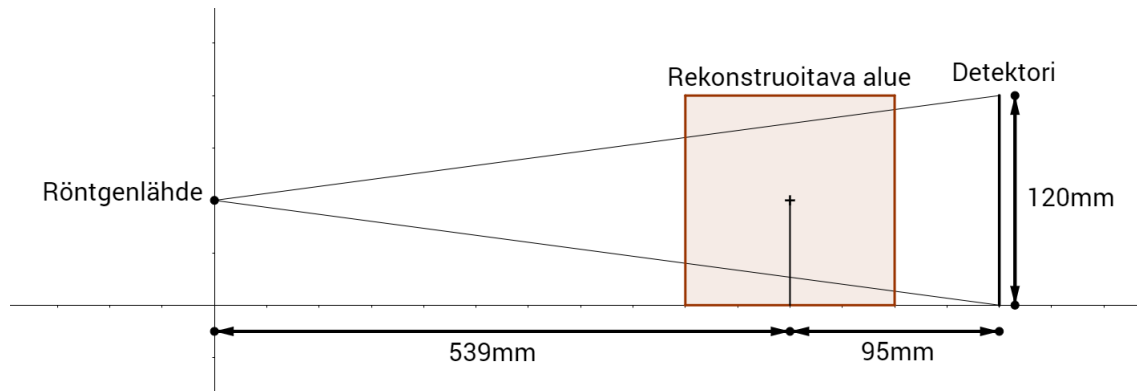
Työn ensimmäinen ongelma oli aineiston kerääminen. 4D-tomografiaa varten tarvitaan valtava määrä röntgendataa liikkuvasta kappaleesta. Käytössä oli Helsingin yliopiston Matematiikan laitoksen Inversio-ongelmien huippututkimusyksikölle keväällä 2015 rakennettu uusi röntgenlaboratorio. Mittalaitteistoon kuului röntgenlähde, röntgendetektor ja pyörivä taso, jonka päälle testikappale asetettiin.

2.1 Mittausjärjestelyt ja aineisto



Kuva 2.1: Testikappaleena käytetty LEGO-robotti

Testikappaleena käytin yksinkertaista LEGO-palikoista rakennettua robottia, jota liikutettiin LEGO Mindstorms -pakettiin kuuluvalla moottorilla. Robottiin kuului kaksi ak-



Kuva 2.2: Röntgenkuvien mittausjärjestelyt sivusta kuvattuna. Röntgenlähteen ja pyörimiskeskusteen etäisyys oli 539mm ja pyörimiskeskusteen ja röntgendetektorin etäisyys oli 95mm.

seleiden ympäri pyörivää kappaletta. Molemmat pyörivät eri nopeuksilla. Robotti asetettiin pyörivän tason päälle, jotta jokaisen robotin liikkeen jälkeen siitä pystyttiin ottamaan röntgenkuva jokaisesta suunnasta.

Mittausjärjestelyssä testikappale asetettiin pyörivälle tasolle, jonka pyörimiskeskuste sijaitsi 95mm etäisyydellä detektorista ja 539mm etäisyydellä röntgenlähteestä. Detektorin koko oli 120mm x 120mm.

Mittaaminen tapahtui siten, että robotin moottoria liikutettiin hieman. Tämä sai aikaan kahden eri akselin ympäri pyörivän kappaleen liikkeen. Kappaleet liikkuvat eri nopeuksilla. Pienen liikkeen jälkeen kappaleesta otettiin täysi röntgenkuvasarja eli 360 röntgenkuvaa kappaleen ympäriltä, asteen välein. Tämän jälkeen robottia liikutettiin uudelleen ja otettiin uudet röntgenkuvat.

Aineistoksi saatiin lopulta kerättyä 360 röntgenkuvaa 69:ltä ajanhetkeltä. Dataa kertyi yhteensä yli 250GB. Röntgendata 17:stä ajanhetkestä jäi kopiointivirheen vuoksi vajaaksi, joten sitä ei käytetä tässä tutkielmassa. Myös neljä ensimmäistä ajanhetkeä jätettiin lopullisesta animaatiosta pois, sillä robotin paristot vaihdettiin neljännen ja viidennen ajanhetken välissä. Tämä liikutti hieman robotin akseleita, mikä aiheutti poikkeaman liikkeessä.

2.2 Ohjelmistot ja rauta

Röntgenlaboratorion laitteisto koostui Oxford Instrumentsin XTF5011 -röntgenputkesta, Hamamatsu Photonicsin C7942CA-22 -röntgendetektorista ja Thorlabsin CR1/M-Z7 -näytteenpyörittimestä. Laitteisto on kytketty ohjaustietokoneeseen, joka kaappasi kuvat

detektorista ja pyöritti kuvattavaa kappaletta halutulla tavalla. Röntgendetektorin kuvankaappaus tehtiin HiPic-ohjelmistolla, Thorlabs tarjoaa oman ohjelmistonsa näytteenpyörittimen ohjaamiseen. Kvanttamista varten kirjoitettiin näytteenpyörittimen ja röntgendetektorin yhdistävä ohjauslogiikka, joka toimi MATLAB-alustan päällä. Tarkemmat tiedot laitteistosta löytyvät röntgenlaboratorion rakentajan, Alexander Meaneyn, Pro Gradu -työstä [7].

Rekonstruktioiden laskeminen tapahtui ASTRA-ohjelmistolla käyttäen sen MATLAB-rajapintaa. ASTRA on ilmainen, vapaan lähdekoodin ohjelmisto, joka on erikoistunut röntgentomografiaan. Se on kehitetty Antwerpenin yliopiston iMinds-VisionLab -laboratoriossa, jonka päätutkimusalue on erilaiset kuvankäsittelyn metodit, joita voidaan hyödyntää esimerkiksi röntgentomografiassa, magneettikuvauksessa ja muodon tunnistuksessa (Shape modeling).

ASTRAan pystyy määrittelemään mittaustilanteen geometrian ja siitä löytyy implementaatiot tässä työssä käytettävistä rekonstruktioalgoritmeista. ASTRA pystyy käyttämään hyväkseen tietokoneen näytönohjainta rekonstruktioita laskettaessa. Näytönohjaimet kykenevät paljon tehokkaampaan laskentaan kuin mihin tietokoneen prosessori pystyy. Näytönohjaimilla tapahtuva laskenta on toteutettu ASTRAssa käyttäen nVidian CUDA-teknologiaa. Rekonstruktiot laskettiin röntgenlaboratorion rekonstruktioiden laskentaa varten rakennetulla tietokoneella.

MATLAB on MathWorks, Inc.:n yleisesti numeerisessa laskennassa käytetty ohjelmistoalusta. MATLABia käytettiin sekä ASTRA:n ohjaamiseksi rekonstruktioita laskettaessa että röntgenlaboratorioiden laitteiden ohjaukseen.

Tulokset visualisoitiin käyttäen Blender-ohjelmistoa. Blender on ilmainen, vapaan lähdekoodin 3D-mallinnusohjelmisto. Blenderiä hallinnoi hollantilainen Blender Foundation ja sillä on satoja kehittäjiä ympäri maailmaa. Visualisaatiot laskettiin tavallisella läppärimellä, yhden framen (640x480) laskeminen kesti n. minuutin ja 10 sekuntia.

Luku 3

Menetelmät

3.1 Matemaattinen perusta

Tämän opinneäytetyön laskenta toteutetaan käyttäen 3D-tomografiaan suunniteltuja matemaattisia menetelmiä eri ajanhetkille. Aikaulottuvuutta ei oteta matemaattisissa malleissa millään tavalla huomioon, tarkoitus on vain tutkia rekonstruktioiden toimivuutta, kun näitä menetelmiä sovelletaan usealle ajanhetkelle.

Röntgentomografia perustuu siihen fysikaaliseen faktaan, että eri alkuaineet vaimentavat röntgensäteilyä eri voimakkuuksilla. Röntgensäde ajatellaan virraksi fotoneita, joista osa vaimenee tai siroaa pois kappaleen läpi kulkiessaan. Lisäksi fotonivirta oletetaan monokromaattiseksi, eli se sisältää vain yhtä valon aallonpituutta. Todellisuudessa röntgenputkella tuotettu säde sisältää useampia aallonpituuksia, jotka voidaan tarvittaessa ottaa huomioon vaimenemismalleissa. Seuraava vaimenemismalli on esitelty T. M. Buzugin kirjassa ([2], s.32-33).

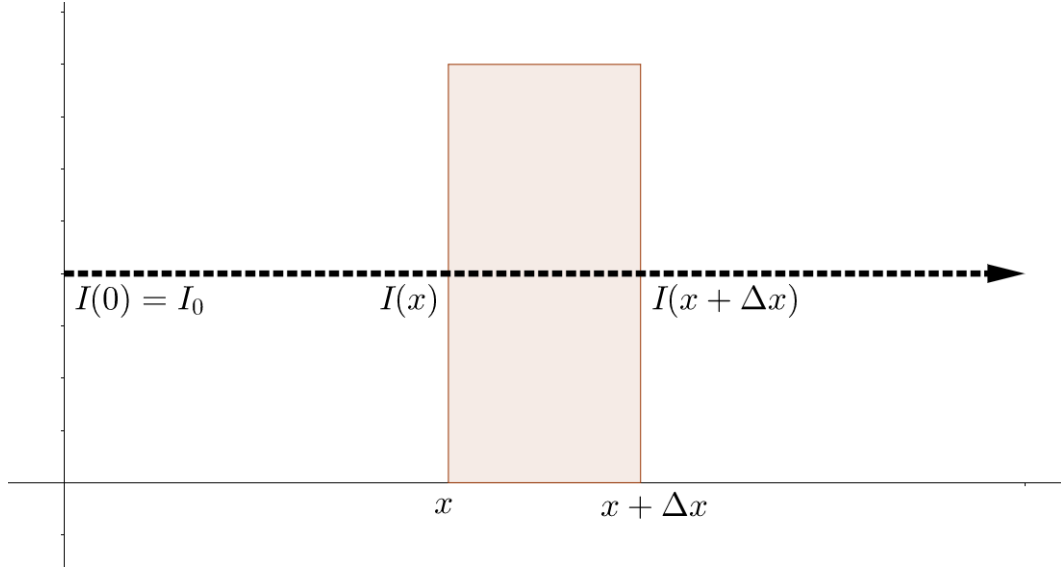
3.2 Lambert-Beerin attenuaatiomalli

Kun röntgensäde kulkee kappaleen läpi, mallinnetaan sen vaimentumista ns. *Lambert-Beerin lailla*, jossa röntgensäteen intensiteetti $I \in \mathbb{R}$ pienenee, kun säde kulkee kappaleen läpi. Intensiteetti on fysikaalinen suure, joka on suoraan verrannollinen fotonien määrään.

Olkoon $I : [0, 1] \rightarrow \mathbb{R}$ funktio, joka kuvaa röntgensäteen intensiteettiä pisteessä x . Kappaleen sisällä kaikilla x pätee

$$(3.1) \quad I(x + \Delta x) = I(x) - f(x)I(x)\Delta x,$$

missä Δx on matka, jonka säde kulkee kappaleen sisällä, ja $f : [0, 1] \rightarrow \mathbb{R}, f(x) \geq 0$ on tutkittavalle kappaleelle ominainen kompaktikantajainen *attenuaatiofunktio*. Funktio



Kuva 3.1: Lambert-Beerin malli visualisoituna. Röntgensäde kulkee kappaleen sisällä matkan Δx ja sen intensiteetti muuttuu Lamber-Beerin mallin mukaisesti.

f kuvaa sitä, kuinka paljon röntgensäteen intensiteetistä häviää kyseisessä pisteessä. Tätä kutsutaan Lambert-Beerin laiksi.

Oletetaan, että kappaleemme on homogeeninen, jolloin funktiolle f pätee $f(x) \equiv \mu, \mu \geq 0$, missä μ on homogeeniselle materiaalille ominainen *attenuaatiokerroin*. Olkoon $I : [0, 1] \rightarrow \mathbb{R}, I(x) > 0$ funktio, joka toteuttaa yhtälön 3.1. Olkoon $I_0 > 0$ röntgensäteiden intensiteetti säteen lähtöpisteessä, siis $I(0) = I_0$. Järjestelemällä yhtälön 3.1 termit uudestaan, saamme

$$I(x + \Delta x) = I(x) - \mu I(x) \Delta x \iff \frac{I(x + \Delta x) - I(x)}{\Delta x} = -\mu I(x)$$

Nyt vasemmalla puolella on intensiteettifunktion erotusosamäärä. Ottamalla puolittain raja-arvon, kun $\Delta x \rightarrow 0$, saamme

$$I'(x) = \lim_{\Delta x \rightarrow 0} \frac{I(x + \Delta x) - I(x)}{\Delta x} = -\mu I(x).$$

Kappaleen sisällä intensiteettifunktio on siis derivoituva ja toteuttaa yo. ensimmäisen

asteen differentiaaliyhtälön. Koska differentiaaliyhtälö on separoituva, saamme helposti

$$\begin{aligned}
 I'(x) = -\mu I(x) &\iff \frac{I'(x)}{I(x)} = -\mu \\
 &\iff \frac{d}{dx} \ln I(x) = -\mu \\
 &\iff \ln I(x) = \int -\mu \, dx \\
 &\iff \ln I(x) = -\mu x + C \\
 &\iff I(x) = e^{-\mu x + C}.
 \end{aligned}$$

Käyttämällä hyväksemme tietoa $I(0) = I_0$, saamme ratkaistua differentiaaliyhtälön

$$I(0) = e^C = I_0,$$

jolloin homogeeniselle kappaleelle saamme määriteltyä Lambert-Beerin attenuaatiomalliin

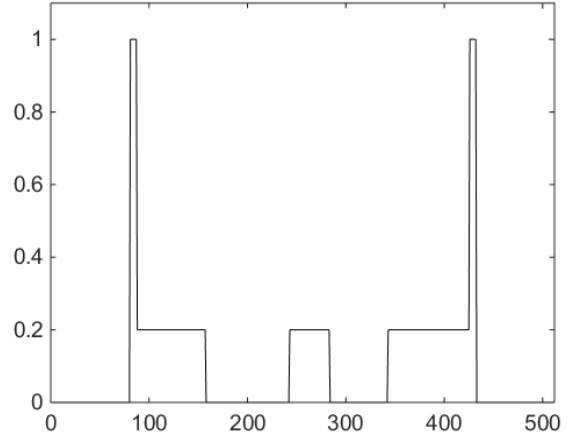
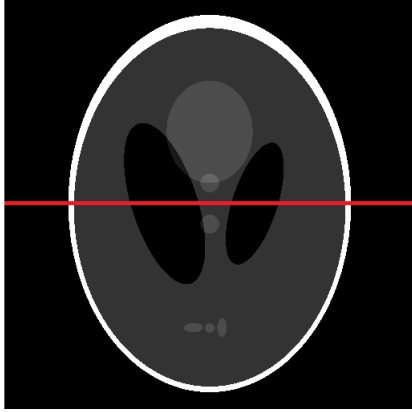
$$(3.2) \quad I(x) = I_0 e^{-\mu x}.$$

3.3 Viivaintegraalit ja niiden mittaaminen

Tutkitaan seuraavaksi kolmiulotteista tilannetta. Olkoon $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ tuntemattoman kappaleen attenuaatiofunktio. Kun röntgensäde kulkee kappaleen f läpi, se vaimenee edellisen mallin perusteella attenuaatiofunktioon f :n rajoittuma säteen kulkemalle suoralle. Osoittautuu, että Lambert-Beerin mallin avulla voimme määritellä kaavan attenuaatiofunktion viivaintegraaleille tämän suoran yli.

Olkoon siis $\gamma \subset \mathbb{R}^3$ suora, joka kulkee röntgenlähteen ja kappaleen läpi. Olkoon x_1 ja x_2 pisteet, jossa röntgensäde osuu kappaleeseen ja tulee ulos kappaleesta. Oletetaan taas, että kappale on homogeeninen eli f on kappaleen sisällä vakiofunktio $f(x) \equiv \mu$. Tällöin f :n integraalille pätee

$$\int_{\gamma} f(x) dx = \mu(x_2 - x_1) = \mu x_2 - \mu x_1.$$



Kuva 3.2: Kuvassa on tomografiassa yleisesti käytetty Shepp-Loganin testikappale. Esimerkkinä attenuaatiofunktio f on testikappaleen 256. rivi

Nyt Lambert-Beerin lain mukaan pätee

$$\begin{aligned}\mu x_2 - \mu x_1 &= \ln \frac{I(x_2)}{I_0} - \ln \frac{I(x_1)}{I_0} \\ &= \ln \frac{I(x_2)}{I_0} - \ln \frac{I_0}{I_0} \\ &= -\ln \frac{I(x_2)}{I_0} \\ &= \ln I_0 - \ln I(x_2).\end{aligned}$$

Siis f :n viivaintegraalille yli γ :n pätee

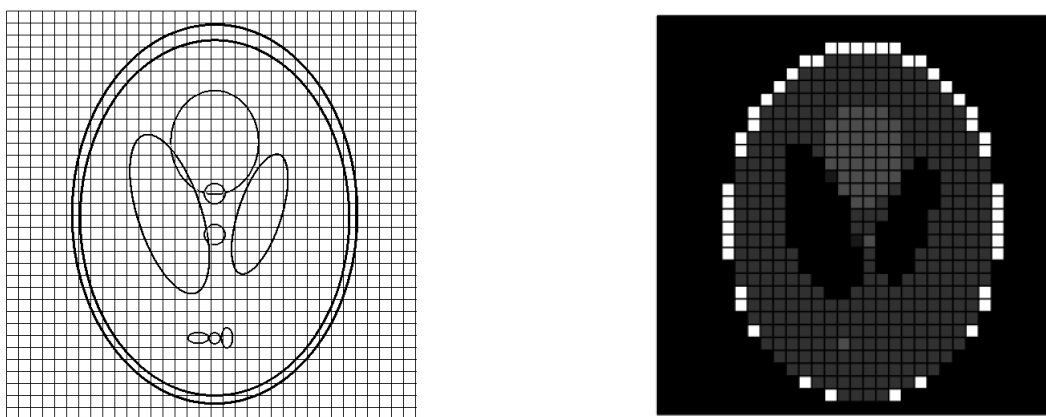
$$(3.3) \quad \int_{\gamma} f(x) dx = \ln I_0 - \ln I_1,$$

missä I_1 on röntgensäteen intensiteetti pisteessä, jossa se poistuu kappaleesta. Sekä I_0 että I_1 voidaan mitata, mikä tekee viivaintegraaleista mahdollisen työkalun funktion f rekonstruointiin.

Viivaintegraalit liittyvät läheisesti niin sanottuun *Radon-muunnokseen*. Radon-muunnoksen kehitti Johann Radon vuonna 1917, hänen tutkiessaan funktioiden esittämistä viivaintegraalien avulla. Funktion f Radon-muunnos kuvaa kyseisen funktion arvoja viivaintegraalien avulla, jotka leikkaavat tutkittavassa pisteessä. Osoittautuu, että nämä integraalit määrittävät funktion arvot yksikäsitteisesti ([1], s. 24-26). Tämä matemaattinen fakta osoittaa sen, että viivaintegraaleja tutkimalla pystymme rekonstruoimaan funktion f .

Monet rekonstruktioimetodit perustuvat Radon-muunnokseen, esimerkiksi kaksiulotteisessa tapauksessa käytetty *Filtered back projection (FBP)* ja kolmiulotteisessa tapauksessa *Feldkamp-Davis-Kress -algoritmi (FDK)*. Tässä opinnäytetyössä keskitytään kuitenkin diskreetteihin, algebrallisiin rekonstruktio menetelmiin.

3.4 Diskretointi ja lineaarinen malli



Kuva 3.3: Yllä on diskretoitu tomografiassa yleisesti käytetty Shepp-Logan -testikappale. Alue Ω on neliö, joka on jaettu pienempiin neliöihin, joita kutsutaan *pikseleiksi*. Kaksiulotteisessa tapauksessa yritetään rekonstruoida näiden pikselien attenuaatioarvoja. Tämän opinnäytetyön tapauksessa rekonstruoidaan kolmiulotteista kappaletta, joten Ω on kuutio, joka jaetaan pienempiin kuutioihin.

Todellisuudessa emme pysty rekonstruoimaan attenuaatiofunktioita f täydellisesti. Mittattu data ei ole koskaan täydellistä ja sisältää mittalaitteista syntynyttä kohinaa. Pystymme ainoastaan rekonstruoimaan diskreetin approksimaation funktiosta f . Jaamme siis funktion f pieniin kuutioihin, joita kutsumme *vokseleiksi*, ja oletamme, että funktio f on vakio jokaisen yksittäisen vokselin sisällä.

Olkoon $\Omega \subset \mathbb{R}^3$ kuutio, jota kutsumme *rekonstruktioalueeksi*. Yritämme rekonstruoida funktion f kuution Ω sisällä. Jaamme tämän kuution $M \in \mathbb{N}$ vokseliin ja merkitsemme

yksittäisten vokselien attenuaatiokertoimia vektorilla

$$\vec{f} = \begin{pmatrix} f_1 \\ \vdots \\ f_M \end{pmatrix} \in \mathbb{R}^M.$$

Asetamme rekonstruoitavan alueen Ω eteen röntgenlähteen ja sen taakse röntgendetektorin ja mittaamme niiden röntgensäteiden γ_j intensiteetin, jotka kulkevat alueen Ω läpi. Röntgendetektorin resoluutiosta riippuen saamme dataa N :stä röntgensäteestä. Olkoon $m_j := \ln I_0 - \ln I_{1,j}$, missä $I_{1,j}$ on detektorin pikselin j arvo. Merkitään näitä arvoja vektorilla

$$\vec{m} = \begin{pmatrix} m_1 \\ \vdots \\ m_N \end{pmatrix} \in \mathbb{R}^N.$$

Kaavan 3.3 perusteella pätee

$$(3.4) \quad m_j = \int_{\gamma_j} f(x) dx = \sum_{i=1}^M a_{ij} f_i,$$

missä f_i on vokselin i attenuaatiokerroin ja $a_{ij} \in \mathbb{R}$ matka, jonka röntgensäde γ_j kulkee vokselissa i . Olkoon nyt A $N \times M$ -matriisi, jolle pätee

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1M} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NM} \end{pmatrix}.$$

Tällöin voimme mallintaa röntgensäteiden attenuaatiota lineaarisella mallilla

$$(3.5) \quad \vec{m} = A\vec{f} + \vec{\epsilon},$$

missä $\vec{\epsilon} \in \mathbb{R}^N$ on kohinatermi.

3.5 Numeeristen menetelmien ongelmat

Edellä esitellyssä mallissa syntyy ongelmia, sillä matriisi A on käytännössä aina erittäin huonokuntainen. Suurin osa alkioista on nollaa, sillä yksi röntgensäde lävistää vain pienen osan koko alueen Ω vokseleista. Tämä tarkoittaa sitä, että käänteismatriisi A^{-1} , mikäli se on olemassa, on herkkä kohinalle ja pienetkin mittausvirheet käytännössä tekevät käänteismatriisilla laskemisen mahdottomaksi.

Yksi tekniikka tutkia matriisin A numeerisia ominaisuuksia on sen *singulaariarvohajotelman* avulla (ks. [1], s. 53-61, [2], s. 207-210).

Lause 3.6. Singular value decomposition: Olkoon A reaallinen $N \times M$ -matriisi. Tällöin on olemassa ortogonaalinen $N \times N$ -matriisi U , ortogonaalinen $M \times M$ -matriisi V sekä diagonaalinen $N \times M$ -matriisi S siten, että

$$(3.7) \quad A = U^T S V = U^T \text{diag}(\sigma_i) V,$$

missä matriisin S diagonaalilla sijaitsevat alkio σ_i ovat matriisin A singulaariarvoja. Matriisin A singulaariarvoille pätee

$$(3.8) \quad \sigma_i = \sqrt{\lambda_i},$$

missä λ_i on matriisin $A^T A$ ominaisarvo.

Tällöin A :n käänteismatriisi, mikäli se on olemassa, saadaan singulaariarvohajotelmas-
ta seuraavalla tavalla

$$(3.9) \quad A^{-1} = (U^T S V)^{-1} = V^T \left(\text{diag} \left(\frac{1}{\sigma_i} \right) \right) U$$

Matriisilla A on käytännössä aina todella pieniä singulaariarvoja, jotka käänteismatriisissa muuttuvat todella suuriksi. Mittauksiin sisältyvä kohina voimistuu käänteismatriisilla laskettaessa todella suureksi, jolloin rekonstruktioista tulee käyttökelvottomia. Yksi ratkaisu tähän on korvata matriisin A liian pienet singulaariarvot nolilla.

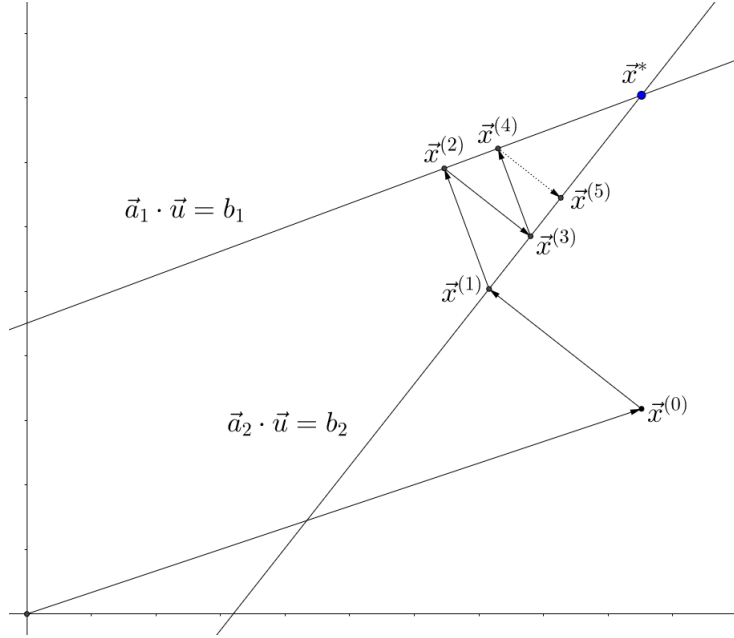
Singulaariarvohajotelman käyttö ei kuitenkaan ole numeerisesti mahdollista todellisille ongelmille. Sen laskeminen on raskas operaatio suurille matriiseille. Tämän opinäytetyön röntgenkuvien resoluutio on 574×560 ja röntgenkuvia on 360 kappaletta ajanhetkeä kohti. Tämä tarkoittaa sitä, että mitattuja röntgensäteitä m_j on yhteensä $N = 574 \cdot 560 \cdot 360 = 115718400$ kappaletta. Lisäksi rekonstruoitavia vokseleita on $M = 512^3 = 134217728$ kappaletta. Tämä tarkoittaa sitä, että matriisi A on kokoluokkaa $10^8 \times 10^8$. Tämän kokoiselle matriisille singulaariarvohajotelman laskeminen on käytännössä mahdotonta. Tämän takia systeemi 3.5 ratkaistaan käyttäen iteratiivisia menetelmiä. Seuraavaksi esitellään kaksi tapaa ratkaista lineaarinen systeemi muotoa

$$(3.10) \quad A\vec{x} = \vec{b},$$

missä A on $N \times M$ -matriisi, $\vec{x} \in \mathbb{R}^M$ ja $\vec{b} \in \mathbb{R}^N$

3.6 Kaczmarzin algoritmi

Puolalainen matemaatikko Stefan Kaczmarz kehitti 1937 iteratiivisen menetelmän ratkaista lineaarisia systeemejä muotoa 3.10. Toinen tässä työssä käytetyistä algoritmeista, SIRT, on approksimaatio Kaczmarzin algoritmista.



Kuva 3.4: Kaczmarzin algoritmin ensimmäiset viisi iteraatiota kahden ulottuvuuden tapauksessa. Kuvassa on esitelty hypertasot (suorat) $\vec{a}_1 \cdot \vec{u} = b_1$ ja $\vec{a}_2 \cdot \vec{u} = b_2$, sekä miten iteraatio poukkoilee näiden välillä.

Olkoon vektori \vec{a}_i matriisin A i :s rivi, ts. $\vec{a}_i = (a_{i1}, \dots, a_{iM})^T \in \mathbb{R}^M$. Tällöin lineaarinen systeemi $A\vec{x} = \vec{b}$ voidaan esittää yhtälöryhmänä

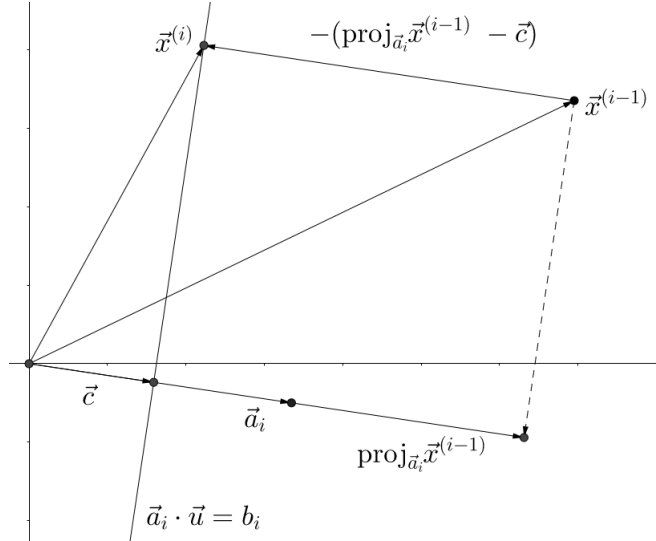
$$(3.11) \quad A\vec{x} = \vec{b} \iff \begin{cases} \vec{a}_1 \cdot \vec{x} &= b_1 \\ &\vdots \\ \vec{a}_N \cdot \vec{x} &= b_N \end{cases}$$

Jokainen näistä yhtälöistä määrittelee *hypertason* M -ulotteisessa euklidisessa avaruudessa. Yhtälöryhmän ratkaisu voidaan ymmärtää näiden hypertasojen leikkauspisteenä avaruudessa \mathbb{R}^M . Systeemin 3.10 ratkaisu siis toteuttaa kaikki yo. yhtälöt.

Algoritmissa lähdetään liikkeelle alkuarvauksesta $\vec{x}^{(0)} \in \mathbb{R}^M$. Tämä vektori projisoidaan ensimmäisen yhtälön määrittelemälle hypertasolle

$$(3.12) \quad P_1 = \{\vec{u} \in \mathbb{R}^M \mid \vec{a}_1 \cdot \vec{u} = b_1\}.$$

Näin saatu piste projisoidaan uudestaan toisen yhtälön määrittelemälle hypertasolle P_2 . Samalla tavalla käydään läpi kaikki hypertasot P_i , kunnes päästään vektoriin $\vec{x}^{(N)}$. Nyt iteraatio aloitetaan uudestaan alkuarvauksena $\vec{x}^{(N)}$ eli tämä vektori projisoidaan taas



Kuva 3.5: Jos avaruutta \mathbb{R}^M tarkastellaan hypertason $\vec{a}_i \cdot \vec{u} = b_i$ suuntaisesti, jakaa taso avaruuden kahteen osaan. Tämä helpottaa huomattavasti projektioiden hahmottamista, ks. [3] kpl 7.2

hypertasolle P_1 . On osoitettu, että tällä tavalla muodostettu vektorijono suppenee kohti systeemin ratkaisua, mikäli sellainen on olemassa ([3] luku 7.2). Voimme näin määritellä iteraation

Lause 3.13. Kaczmarzin algoritmi: Olkoon $\vec{x}_0 \in \mathbb{R}^M$. Tällöin iteraatio

$$(3.14) \quad \vec{x}^{(i)} = \vec{x}^{(i-1)} - \frac{\vec{a}_i \cdot \vec{x}^{(i-1)} - b_i}{\|\vec{a}_i\|^2} \vec{a}_i.$$

suppenee kohti lineaarisen systeemin $A\vec{x} = \vec{b}$ ratkaisua, jos sellainen on olemassa.

Todistetaan vielä iteraation kaava. Haluamme siis laskea edellisen iteraation tuloksen $\vec{x}^{(i-1)}$ projektion hypertasolle P_i . Hypertasoa P_i voidaan ajatella suorana kaksiulotteisessa koordinaatistossa, tavoitteenamme on siis etsiä pisteen kohtisuora projektio tälle suoralle (ks. kuva 3.6).

Etsitään aluksi vektorin \vec{a} suuntainen vektori \vec{c} , joka kuuluu hypertasolle P_i . Tällaiselle vektorille pätee $\vec{c} = c\vec{a}_i$, $c \in \mathbb{R}$ ja koska vektori \vec{c} sijaitsee hypertasolla, se toteuttaa yhtälön

3.12. Tällöin pätee

$$\begin{aligned}\vec{a}_i \cdot \vec{c} &= b_i \\ \iff c(\vec{a}_i \cdot \vec{a}_i) &= b_i \\ \iff c\|\vec{a}_i\|^2 &= b_i \\ \iff c &= \frac{b_i}{\|\vec{a}_i\|^2}.\end{aligned}$$

Lasketaan nyt edellisen iteraation tuloksen $\vec{x}^{(i-1)}$ projektio uudelle hypertasolle. Vektorin $\vec{x}^{(i-1)}$ projektio vektorille \vec{a}_i on

$$\text{proj}_{\vec{a}_i} \vec{x}^{(i-1)} = \frac{\vec{a}_i \cdot \vec{x}^{(i-1)}}{\|\vec{a}_i\|^2} \vec{a}_i,$$

joten haluamamme vektori on

$$\begin{aligned}\vec{x}^{(i)} &= \vec{x}^{(i-1)} - (\text{proj}_{\vec{a}_i} \vec{x}^{(i-1)} - \vec{c}) \\ &= \vec{x}^{(i-1)} - \left(\frac{\vec{a}_i \cdot \vec{x}^{(i-1)}}{\|\vec{a}_i\|^2} \vec{a}_i - \frac{b_i}{\|\vec{a}_i\|^2} \vec{a}_i \right) \\ &= \vec{x}^{(i-1)} - \frac{\vec{a}_i \cdot \vec{x}^{(i-1)} - b_i}{\|\vec{a}_i\|^2} \vec{a}_i,\end{aligned}$$

mikä onkin haluttu iteraatiokaava 3.14.

Edellisessä todistuksessa oletettiin, että lineaarisen systeemin 3.10 muodostamilla hypertasoilla todellakin on yhteinen leikkauspiste. Tämä tarkoittaa sitä, että matriisilla A on yhtä paljon rivejä kuin sarakkeita, siis $N = M$. Lisäksi tehtiin oletus, että kohina-termi $\vec{\epsilon} = \vec{0}$, eli mittausdata ei sisällä yhtään kohinaa. Todellisessa tapauksessa aidolla mittausdatalla laskettaessa kuitenkin kumpikaan näistä oletuksista ei pidä paikkansa.

Tässä työssä mitattuja röntgensäteitä on enemmän, kuin mitä rekonstruoitavia vokseleita. Siis pätee $M > N$. Tällöin lineaarisen systeemin 3.10 muodostamilla hypertasoilla ei löydy yhteistä leikkauspistettä. Tällaisissa tapauksissa algoritmi jää oskilloimaan eri hypertasojen leikkauspisteiden läheisyyteen ([3] s.281). Näiden vektorien tulisi kuitenkin olla hyviä approksimaatioita todelliselle kappaleelle.

3.7 ART ja SIRT

ART eli *Algebraic reconstruction technique* on algoritmi, jota Hounsfield käytti 1970-luvulla ensimmäisiä röntgenrekonstruktioita laskiessaan. ART perustuu Kaczmarzin algoritmiin, mutta Hounsfield kehitti ratkaisun itsenäisesti tietämättä Kaczmarzin työstä ([2], s.211).

Projektio hypertasolle P_i voidaan tulkita yhden röntgensäteen b_i mittauksen hyödyntämiseksi. Tällä tulkinnalla päivitämme jokaisella iteraatiolla yhden röntgensäteen sisältämän information rekonstruktioon. Koska päivitämme vain yhden röntgensäteen vaikutuksen kerrallaan, ovat iteraatiot riippumattomia toisistaan. Tästä syystä pystymme päivittämään vain yksittäisiä vokseleita kerrallaan. Kaavasta 3.14 saamme yksittäisen vokselin attenuaatioarvoksi

$$(3.15) \quad x_j^{(i)} = x_j^{(i-1)} + \frac{b_i - q_i}{\|\vec{a}_i\|^2} \vec{a}_{ij},$$

missä

$$\begin{aligned} q_i &= \vec{x}^{(i-1)} \cdot \vec{a}_i \\ &= \sum_{k=1}^N x_k^{(i-1)} a_{ik}. \end{aligned}$$

Tällöin jokaisessa iteraatiossa päivitämme yksittäisiä vokseleita arvolla

$$(3.16) \quad \Delta x_j^{(i)} = \vec{x}_j^{(i)} - \vec{x}_j^{(i-1)} = \frac{b_i - q_i}{\|\vec{a}_i\|^2} \vec{a}_{ij}.$$

ARTista on muutamia eri implementaatioita, joissa käytetään kaavalle 3.16 erilaisia approksimaatioita. Yksi helpoimmista on korvata vektorien a_i arvot ykkösillä tai nolilla riippuen siitä, kuinka pitkän matkan röntgensäde i kulkee eri vokseleissa. Käyttäisimme siis kaavan 3.16 sijaan kaavaa

$$(3.17) \quad \Delta x_j^{(i)} = \frac{b_i - q_i}{N_i},$$

missä N_i on niiden vokselien lukumäärä, jotka röntgensäde i lävistää. Tämä ei välttämättä kuitenkaan ole hyvä approksimaatio $\|\vec{a}_i\|^2$:lle, mikä saattaa aiheuttaa artefakteja rekonstruktioissa. Parempi approksimaatio kaavalle 3.16 on

$$(3.18) \quad \Delta x_j^{(i)} = \frac{b_i}{L_i} - \frac{q_i}{N_i},$$

missä L_i on matka, jonka röntgensäde i kulkee rekonstruoitavalla alueella.

SIRT eli *Simultaneous iterative reconstruction technique* on muunnelmä ARTista. Vokseleita päivitetään käyttämällä kaavoja 3.17 ja 3.18, mutta vokselien arvot päivitetään vasta lopuksi, kun kaikki röntgensäteet i on käyty läpi. Päivitetty arvo on keskiarvo kaikista vokselin päivityksistä ([3], s. 284).

$$\Delta \hat{x}_j^{(i)} = \frac{1}{N} \sum_{k=1}^N \Delta x_j^{(k)}.$$

SIRT on toinen tässä työssä käytetyistä algoritmeista.

3.8 Kvadraattiset rekonstruktio menetelmät

Toinen tapa ratkaista lineaarisia systeemejä muotoa $A\vec{x} = \vec{b}$ perustuu siihen matemaatiseen faktaan, että tämäntyyppisten yhtälöiden ratkaisut ovat erään tunnetun, hyvin käyttäytyvän kvadraattisen funktion f globaaleja minimejä. Muokkaamalla yo. yhtälöä hieman, saamme

$$A\vec{x} = \vec{b} \iff A\vec{x} - \vec{b} = \vec{0}.$$

Olkoon nyt $g(\vec{x}) = A\vec{x} - \vec{b}$. Funktio g osoittautuu funktion f gradientiksi. Koska etsimme vektoria \vec{x} , jossa funktion f gradientti g on nolla, etsimme siis oikeastaan funktion f minimi- tai maksimikohtia. Käy ilmi, että funktiolla f on vain yksi globaali minimi, jonka on siis oltava lineaarisen systeemin 3.10 ratkaisu. Tarvitsemme vain iteratiivisen algoritmin, joka minimoi funktion f .

Määritelmä 3.19. Olkoon A symmetrinen $N \times N$ -matriisi ja $\vec{u} \in \mathbb{R}^N$. Tällöin A on *positiivisesti definiitti*, jos kaikille vektoreille $\vec{u} \neq \vec{0}$ pätee

$$\vec{u}^T A \vec{u} > 0.$$

Lause 3.20. Olkoon A symmetrinen, *positiivisesti definiitti* $N \times N$ -matriisi, $\vec{b} \in \mathbb{R}^N$. Tällöin lineaarisen systeemin $A\vec{x} = \vec{b}$ ratkaisu on vektori \vec{x} , joka minimoi funktion

$$f : \mathbb{R}^N \rightarrow \mathbb{R}, f(\vec{x}) = \frac{1}{2} \vec{x}^T A \vec{x} - \vec{x}^T \vec{b}.$$

Todistetaan ylläoleva lause. Osoitetaan aluksi, että funktio g todellakin on funktion f gradientti. Lasketaan ensimmäiseksi auki funktion f esitystapa useamman muuttujan suhteen.

$$\begin{aligned} f(\vec{x}) &= \frac{1}{2} \vec{x}^T A \vec{x} - \vec{x}^T \vec{b} = \frac{1}{2} (\vec{x} \cdot A\vec{x}) - \vec{x} \cdot \vec{b} \\ &= \frac{1}{2} \sum_{i=1}^N x_i \sum_{j=1}^N a_{ij} x_j - \sum_{i=1}^N x_i b_i \end{aligned}$$

Lasketaan seuraavaksi funktion f osittaisderivaatta komponentin x_k suhteen. Suoralla

summien osittaisderivoinnilla ja matriisin A symmetrisyyden nojalla saamme

$$\begin{aligned}\partial_k f(x) &= \frac{1}{2} \left(\sum_{i=1}^N \partial_k x_i \sum_{j=1}^N a_{ij} x_j \right) - \sum_{i=1}^N \partial_k x_i b_i \\ &= \frac{1}{2} (a_{1k} x_1 + \cdots + (a_{k1} x_1 + \cdots + 2a_{kk} x_k + \cdots + a_{kN} x_N) + \cdots + a_{Nk} x_N) - b_k \\ &= \frac{1}{2} (2a_{1k} x_1 + \cdots + 2a_{Nk} x_N) - b_k \\ &= \vec{a}_k \cdot \vec{x} - b_k,\end{aligned}$$

missä vektori \vec{a}_k on matriisin A k :s sarake. Koska osittaisderivaatat ovat selvästi jatkuvia kaikilla k ja matriisi A on symmetrinen, on funktion f gradientti

$$\nabla f(\vec{x}) = \begin{pmatrix} \vec{a}_1 \cdot \vec{x} - b_1 \\ \vdots \\ \vec{a}_N \cdot \vec{x} - b_N \end{pmatrix} = \begin{pmatrix} \vec{a}_1 \cdot \vec{x} \\ \vdots \\ \vec{a}_N \cdot \vec{x} \end{pmatrix} - \vec{b} = A\vec{x} - \vec{b} = g(\vec{x}).$$

Osoitetaan vielä, että funktiolla f on yksikäsitteinen, globaali minimi. Tehdään tämä osoittamalla ensin, että matriisi A on kääntyvä, jolloin funktiolla f on olemassa yksikäsitteinen ääriarvokohta $A^{-1}\vec{b}$. Osoitetaan vielä tämän jälkeen, että ääriarvokohdan on oltava globaali minimi.

Matriisi A on kääntyvä, jos sen ydin sisältää vain nollavektorin. Tehdään vastaoletus: Oletetaan, että on olemassa vektori $\vec{z} \neq \vec{0}$, jolle pätee $A\vec{z} = \vec{0}$. Tällöin pätee myös, että

$$\vec{z}^T A \vec{z} = \vec{z}^T \vec{0} = 0,$$

mikä on ristiriita, sillä matriisi A on positiivisesti definiitti. Siis matriisin A on oltava kääntyvä. Ratkaistaan nyt funktion f ääriarvokohdat gradientin nollakohdista:

$$g(\vec{x}) = \vec{0} \iff A\vec{x} - \vec{b} = \vec{0} \iff \vec{x} = A^{-1}\vec{b}.$$

Koska funktio f on määritelty koko avaruudessa \mathbb{R}^M , löytyvät ääriarvokohdat pelkästään gradientin nollakohdista. Siis funktiolla f on vain yksi ääriarvokohta pisteessä $A^{-1}\vec{b}$.

Osoitetaan vielä, että funktio f saa kyseisessä ääriarvokohdassa miniminsä. Lauseen 3.6 mukaan reaalisella matriisilla on olemassa singulaariarvohajotelma $A = U^T S V$. Lisäksi, koska matriisi A on symmetrinen, pätee myös

$$A = A^T \iff U^T S V = V^T S U \iff U = V.$$

Käyttämällä singulaariarvohajotelmaa, löydämme funktiolle f eräällä tavalla paremman esitystavan, jolla voimme todistaa pisteen $A^{-1}\vec{b}$ todella olevan funktion f minimikohta.

Todetaan aluksi, että koska matriisi U on ortogonaalinen, löytyy yksikäsitteinen vektori $\vec{c} \in \mathbb{R}^N$ siten, että $\vec{b} = U^T \vec{c}$. Nyt huomaamme, että

$$\begin{aligned} f(\vec{x}) &= \frac{1}{2} \vec{x}^T A \vec{x} - \vec{x}^T \vec{b} \\ &= \frac{1}{2} \vec{x}^T U^T S U \vec{x} - \vec{x}^T \vec{b} \\ &= \frac{1}{2} \vec{x}^T U^T S U \vec{x} - \vec{x}^T U^T \vec{c} \\ &= \frac{1}{2} (U \vec{x})^T S U \vec{x} - (U \vec{x})^T \vec{c}. \end{aligned}$$

Olkoon nyt $\vec{y} = U \vec{x}$. Määritellään nyt funktio h seuraavasti

$$(3.21) \quad h(\vec{y}) := \frac{1}{2} \vec{y}^T S \vec{y} - \vec{y}^T \vec{c}.$$

Sen sijaan, että etsimme suoraan funktion f minimiä, voimme etsiä funktion h minimiä avaruudessa, jota on käännetty matriisilla U . Funktion h minimointi osoittautuu huomattavasti helpommaksi, sillä matriisi S on diagonaalinen.

Funktiot f ja h ovat lähes identtisiä. Matriisin S on oltava myös positiivisesti definiitti, sillä matriisin A ominaisarvot ovat kaikki positiivisia. Jos ominaisvektorilla $\vec{z} \neq \vec{0}$ on ominaisarvo $\lambda \in \mathbb{R}$, niin tällöin matriisin A positiividefiniittisyyden perusteella

$$\vec{z}^T A \vec{z} = \lambda \vec{z}^T \vec{z} > 0 \iff \lambda > 0.$$

Matriisin S diagonaalilla olevat singulaariarvot s_i ovat ominaisarvojen neliöjuuria, joten tällöin myös niiden täytyy olla positiivisia. Tällöin mielivaltaiselle vektorille $\vec{z} \neq \vec{0}$ pätee

$$\vec{z}^T S \vec{z} = \sum_{i=1}^N s_i z_i^2 > 0.$$

Matriisilla S on positiividefiniittisyyden lisäksi se hyvä ominaisuus, että se on diagonaalinen. Tekemällä funktiolle h vastaavat päättelyt kuin funktiolle f , voimme todeta, että funktiolla h on yksikäsitteinen ääriarvokohta pisteessä $\vec{y} = S^{-1} \vec{c}$. Nyt, jos funktiolla h on minimikohta tässä pisteessä, niin tällöin funktiolla f on oltava minimikohta pisteessä

$$U^T (S^{-1} \vec{c}) = U^T S^{-1} U \vec{b} = A^{-1} \vec{b},$$

sillä $f(\vec{x}) = h(U \vec{x})$. Meidän täytyy siis enää osoittaa, että funktiolla h on minimikohta pisteessä $S^{-1} \vec{c}$. Tehdään tämä tutkimalla funktion h suunnattua derivaattaa vektorin

$\vec{v} \in \mathbb{R}^N$ suuntaan kyseisessä pisteessä. Olkoon $t \in \mathbb{R}$. Merkitään funktion h suunnattua derivaattaa yo. pisteessä funktiolla

$$\gamma_{\vec{v}}(t) := h(S^{-1}\vec{c} + t\vec{v}).$$

Nyt, koska matriisi S on diagonaalinen, pätee $(S^{-1})^T = S^{-1}$, jolloin

$$\begin{aligned} \gamma'_{\vec{v}}(t) &= \frac{d}{dt} \left(\frac{1}{2} (S^{-1}\vec{c} + t\vec{v})^T S (S^{-1}\vec{c} + t\vec{v}) - (S^{-1}\vec{c} + t\vec{v})^T \vec{c} \right) \\ &= \frac{d}{dt} \left(\frac{1}{2} (\vec{c}^T (S^{-1})^T + t\vec{v}^T) (\vec{c} + tS\vec{v}) - (\vec{c}^T (S^{-1})^T + t\vec{v}^T) \vec{c} \right) \\ &= \frac{d}{dt} \left(\frac{1}{2} \vec{c}^T (S^{-1})^T \vec{c} + \frac{t}{2} \vec{c}^T (S^{-1})^T S \vec{v} + \frac{t}{2} \vec{v}^T \vec{c} + \frac{t^2}{2} \vec{v}^T S \vec{v} - \vec{c}^T (S^{-1})^T \vec{c} - t\vec{v}^T \vec{c} \right) \\ &= \frac{d}{dt} \left(\frac{t^2}{2} \vec{v}^T S \vec{v} - \frac{1}{2} \vec{c}^T (S^{-1})^T \vec{c} + \frac{t}{2} \vec{c}^T (S^{-1})^T S \vec{v} - \frac{t}{2} \vec{v}^T \vec{c} \right) \\ &= \frac{d}{dt} \left(\frac{t^2}{2} \vec{v}^T S \vec{v} - \frac{1}{2} \vec{c}^T (S^{-1})^T \vec{c} \right) \\ &= t\vec{v}^T S \vec{v}. \end{aligned}$$

Nyt, koska matriisi S on positiivisesti definiitti, pätee $\vec{v}^T S \vec{v} > 0$, joten suunnatulla derivaatalla on nollakohta pisteessä $t = 0$. Koska suuntavektori \vec{v} oli mielivaltainen, on pisteen $s^{-1}\vec{c}$ oltava funktion h minimikohta. Siis funktiolla f on oltava yksikäsitteinen minimikohta pisteessä $A^{-1}\vec{b}$.

Aikojen saatossa on kehitetty erilaisia menetelmiä minimoida funktiota f . Esimerkiksi Cauchyn vuonna 1847 kehittämä *Gradient descent*-menetelmä ja sen erilaiset variaatiot ovat yleisesti käytössä. Seuraavaksi esiteltävä CGLS-algoritmi on saman tyyppinen metodi, joka perustuu *konjugoituihin* vektoreihin.

3.9 CGLS

Minimoitaessa kvadraattista funktiota, haluamme löytää iteratiivisen menetelmän, joka suppenee kohti systeemin ratkaisua. Haluamme siis löytää iteraation

$$(3.22) \quad \vec{x}_{k+1} = \vec{x}_k + \alpha_k \vec{d}_k,$$

missä vektorit \vec{d}_k ovat nk. *etsintäsuuntia* (*search direction*) ja kertoimet $\alpha_k \in \mathbb{R}$ ovat *askelpituuksia* (*step length*), jotka kertovat, kuinka pitkälle suuntaan \vec{d}_k kuljemme. Lisäksi tulisi päteä, että $\vec{x}_k \rightarrow \vec{x}^*$, kun $k \rightarrow \infty$, missä \vec{x}^* on lineaarisen systeemin 3.10 ratkaisu. Yksi tällainen menetelmä löytyy käyttämällä hyväksemme konjugoituja vektoreita.

Määritelmä 3.23. Olkoon A symmetrinen, positiivisesti definiitti $N \times N$ -matriisi ja $\vec{u}, \vec{v} \in \mathbb{R}^N$. Tällöin vektorit \vec{u} ja \vec{v} ovat *konjugoituja* matriisiin A suhteen tai A -ortogonaalisia, jos

$$\vec{u}^T A \vec{v} = 0$$

Olkoon $D = \{\vec{d}_1, \dots, \vec{d}_N\}, \vec{d}_i \in \mathbb{R}^N$ joukko keskenään A -ortogonaalisia vektoreita. Osoittautuu, että tällainen joukko muodostaa kannan avaruudelle \mathbb{R}^N . Tällöin voimme ilmaista lineaarisen systeemin 3.10 ratkaisun vektorien \vec{d}_i avulla. Osoittautuu myös, että tarvittavien kertoimien α_i määrittäminen on helppoa A -ortogonaalisille vektoreille.

Todistetaan seuraavaksi edelliset väittämät. Seuraavat todistukset löytyvät Muellerin ja Siltasen kirjasta [1] sivuilta 79-81.

Lause 3.24. *Olkoon $D = \{\vec{d}_1, \dots, \vec{d}_N\}, \vec{d}_i \in \mathbb{R}^N$ joukko keskenään A -ortogonaalisia vektoreita. Tällöin ne muodostavat kannan avaruudelle \mathbb{R}^N .*

Todistetaan yo. lause. Olkoon $\alpha_i \in \mathbb{R}$ sellaisia, että

$$\alpha_1 \vec{d}_1 + \dots + \alpha_N \vec{d}_N = \vec{0}.$$

Tällöin kertomalla yo. yhtälöä vasemmalta ensin matriisilla A ja sitten vektorilla $\vec{d}_j^T, j \in \{1, \dots, N\}$, saamme

$$\begin{aligned} \alpha_1 \vec{d}_1 + \dots + \alpha_N \vec{d}_N &= \vec{0} \\ \iff \alpha_1 \vec{d}_1^T A \vec{d}_1 + \dots + \alpha_N \vec{d}_j^T A \vec{d}_N &= 0. \end{aligned}$$

Nyt vektorien \vec{d}_i A -ortogonaalisuuden perusteella $\vec{d}_j^T A \vec{d}_i = 0$ kaikilla $i \neq j$. Matriisi A on positiivisesti semidefiniitti, joten $\vec{d}_j^T A \vec{d}_j > 0$. Tällöin

$$\alpha_j \vec{d}_j^T A \vec{d}_j = 0 \implies \alpha_j = 0.$$

Koska j oli mielivaltainen, pätee $\alpha_j = 0$ kaikilla $j \in \{1, \dots, N\}$, siis vektorit \vec{d}_i ovat keskenään lineaarisesti riippumattomia ja muodostavat siten kannan avaruudelle \mathbb{R}^N . Todistetaan seuraavaksi iteraatiokaava lineaarisen systeemin 3.10 ratkaisun laskemiseksi, eli niin sanottu *Conjugate direction theorem*.

Lause 3.25. Conjugate direction theorem: *Olkoon A symmetrinen, positiividefiniittinen $N \times N$ -matriisi. Olkoon $D = \{\vec{d}_0, \dots, \vec{d}_{N-1}\}$ joukko A -ortogonaalisia vektoreita. Tällöin mille tahansa alkuarvaukselle $\vec{x}_0 \in \mathbb{R}^N$ pätee, että iteraatio*

$$(3.26) \quad \vec{x}_{k+1} = \vec{x}_k + \alpha_k \vec{d}_k,$$

missä

$$(3.27) \quad \alpha_k = -\frac{\vec{g}_k^T \vec{d}_k}{\vec{d}_k^T A \vec{d}_k} \quad \text{ja} \quad \vec{g}_k = A\vec{x}_k - \vec{b},$$

suppenee lineaarisen systeemin $A\vec{x} = \vec{b}$ ratkaisuun \vec{x}^* N :llä iteraatiolla, siis $\vec{x}_N = \vec{x}^*$.

Todistetaan yo. lause. Lineaarisella systeemillä $A\vec{x} = \vec{b}$ on olemassa yksikäsitteinen ratkaisu, sillä matriisi A on positiividefiniittisenä matriisina kääntyvä. Koska vektorit \vec{d}_i ovat avaruuden \mathbb{R}^N kanta, voimme esittää ratkaisun \vec{x}^* ja alkuarvauksen x_0 erotuksen vektorien \vec{d}_i lineaarikombinaationa:

$$\vec{x}^* - \vec{x}_0 = \alpha_0 \vec{d}_0 + \cdots + \alpha_{N-1} \vec{d}_{N-1} = \sum_{i=0}^{N-1} \alpha_i \vec{d}_i.$$

Kertomalla yo. esitystä vasemmalta termillä $\vec{d}_k^T A$, $k \in \{0, \dots, N-1\}$ saamme vektoreiden \vec{d}_i A -ortogonaalisuuden perusteella

$$\begin{aligned} \vec{d}_k^T A(\vec{x}^* - \vec{x}_0) &= \sum_{i=0}^{N-1} \alpha_i \vec{d}_k^T A \vec{d}_i = \alpha_k \vec{d}_k^T A \vec{d}_k \\ \implies \alpha_k &= \frac{\vec{d}_k^T A(\vec{x}^* - \vec{x}_0)}{\vec{d}_k^T A \vec{d}_k} \end{aligned}$$

Iteraation 3.26 perusteella tiedämme, että

$$\vec{x}_k = \vec{x}_0 + \sum_{i=0}^{k-1} \alpha_i \vec{d}_i,$$

joten kertomalla yo. yhtälöä taas vasemmalta termillä $\vec{d}_k^T A$ saamme A -ortogonaalisuuden perusteella

$$\begin{aligned} \vec{d}_k^T A(\vec{x}_k - \vec{x}_0) &= \sum_{i=0}^{k-1} \alpha_i \vec{d}_k^T A \vec{d}_i = \vec{0} \\ \implies \vec{d}_k^T A \vec{x}_k &= \vec{d}_k^T A \vec{x}_0. \end{aligned}$$

Sijoittamalla tämän yo. kaavaan kertoimille α_k saamme

$$\begin{aligned}
\alpha_k &= \frac{\vec{d}_k^T A \vec{x}^* - \vec{d}_k^T A \vec{x}_k}{\vec{d}_k^T A \vec{d}_k} \\
&= \frac{\vec{d}_k^T \vec{b} - \vec{d}_k^T A \vec{x}_k}{\vec{d}_k^T A \vec{d}_k} \\
&= \frac{-\vec{d}_k^T (A \vec{x}_k - \vec{b})}{\vec{d}_k^T A \vec{d}_k} \\
&= -\frac{\vec{d}_k^T \vec{g}_k}{\vec{d}_k^T A \vec{d}_k},
\end{aligned}$$

mikä todistaa iteraation 3.26.

Ongelmaksi jää, miten löydämme vektorit \vec{d}_i . Osoittautuu, että pystymme muodostamaan A -konjugoidun vektorijoukon D nk. *Gramm-Schmidtin konjugoinnilla* joukosta $U = \{\vec{u}_0, \dots, \vec{u}_{N-1}\}$ lineaarisesti riippumattomia vektoreita. Seuraavat todistukset löytyvät J. R. Schewchukin kirjoittamasta CGLS-esittelystä [5].

Lause 3.28. *Gram-Schmidtin konjugointi:* Olkoon $U = \{\vec{u}_0, \dots, \vec{u}_{N-1}\}$ joukko lineaarisesti riippumattomia vektoreita. Tällöin vektorit \vec{d}_i

$$(3.29) \quad \vec{d}_i = \vec{u}_i + \sum_{k=0}^{i-1} \beta_{ik} \vec{d}_k,$$

missä

$$(3.30) \quad \beta_{ik} = -\frac{\vec{u}_i^T A \vec{d}_k}{\vec{d}_k^T A \vec{d}_k}$$

ovat A -konjugaatteja.

Todistetaan yo. lause induktiolla. Ideana on valita aluksi $\vec{d}_0 = \vec{u}_0$, minkä jälkeen iteraatiivisesti valitaan aina uusi lineaarisesti riippumaton vektori \vec{u}_i , mistä vähennetään kaikki komponentit, jotka eivät ole A -konjugaatteja edellisten vektorien \vec{d}_i kanssa. Haluamme siis määrittää kertoimet β_{ik} siten, että pätee $\vec{d}_j^T A \vec{d}_i = 0$ kaikilla $i \neq j$.

Määritetään aluksi kertoimet β_{ik} . Käytetään tähän näppärää laskutoimitusta. Oletetaan, että vektorit \vec{d}_i ovat keskenään A -konjugaatteja. Tällöin kertomalla yhtälöä 3.29

termillä $\vec{d}_j^T A$ vasemmalta saamme vektorien \vec{d}_i A -konjugaation perusteella

$$\begin{aligned}\vec{d}_j^T A \vec{d}_i &= \vec{d}_j^T A \left(\vec{u}_i + \sum_{k=0}^{i-1} \beta_{ik} \vec{d}_k \right) \\ 0 &= \vec{d}_j^T A \vec{u}_i + \sum_{k=0}^{i-1} \beta_{ik} \vec{d}_j^T A \vec{d}_k \\ 0 &= \vec{d}_j^T A \vec{u}_i + \beta_{ij} \vec{d}_j^T A \vec{d}_j \\ \beta_{ij} &= -\frac{\vec{d}_j^T A \vec{u}_i}{\vec{d}_j^T A \vec{d}_j}\end{aligned}$$

Kertoimien β_{ik} täytyy siis olla muotoa 3.30. Oletetaan nyt, että vektorit $\vec{d}_k, k \in \{0, \dots, i-1\}$ ovat A -konjugaatteja keskenään. Osoitetaan, että tällöin vektori \vec{d}_i on myös A -konjugaatti kaikkien näiden vektorien kanssa. Olkoon $j < i$. Tällöin kertomalla yhtälöä 3.29 termillä $\vec{d}_j^T A$ vasemmalta saamme

$$\vec{d}_j^T A \vec{d}_i = \vec{d}_j^T A \left(\vec{u}_i + \sum_{k=0}^{i-1} \beta_{ik} \vec{d}_k \right) = \vec{d}_j^T A \vec{u}_i + \sum_{k=0}^{i-1} \beta_{ik} \vec{d}_j^T A \vec{d}_k = \vec{d}_j^T A \vec{u}_i - \frac{\vec{d}_j^T A \vec{u}_i}{\vec{d}_j^T A \vec{d}_j} \vec{d}_j^T A \vec{d}_j = 0.$$

Siis iteraation seuraava vektori on A -konjugaatti kaikkien edellisten vektorien kanssa. Koska A -ortogonaalisuus on vaihdannainen relaatio, on vektori \vec{d}_i A -ortogonaalinen myös vektorin $\vec{d}_j, j > i$ kanssa kaikilla i .

Meillä on nyt matemaattinen koneisto, jolla pystymme tuottamaan algoritmin, joka suppenee kohti lineaarisen systeemin 3.10 ratkaisua. Meidän tarvitsee vain valita N kapaleutta lineaarisesti riippumattomia vektoreita (esimerkiksi koordinaattiakselit), suorittaa Gram-Schmidtin konjugointi ja käyttää iteraatiota lauseessa 3.25. Ongelmaksi muodostuu, että Gram-Schmidtin konjugointi on aikavaativuudeltaan luokkaa $O(N^3)$ ja joudumme pitämään kaikki vektorit \vec{d}_i muistissa pystyäksemme laskemaan vektorin \vec{d}_{i+1} . CGLS korjaa tämän ongelman näppärästi valitsemalla konjugoitavat vektorit \vec{u}_i jokaisen iteraation gradientiksi. Tästä seuraa nimitys *conjugate gradient*.

Valitaan siis $\vec{u}_i = \vec{g}_i := A\vec{x}_i - \vec{b}$. Tällöin Gram-Schmidtin konjugoinnin kertoimille β_{ik} pätee:

$$\beta_{ik} = -\frac{\vec{u}_i^T A \vec{d}_k}{\vec{d}_k^T A \vec{d}_k} = -\frac{\vec{g}_i^T A \vec{d}_k}{\vec{d}_k^T A \vec{d}_k}.$$

Huomataan myös, että gradienteille \vec{g}_k pätee

$$\begin{aligned}\vec{g}_{k+1} &= A\vec{x}_{k+1} - \vec{b} \\ &= A(\vec{x}_k + \alpha_k \vec{d}_k) - \vec{b} \\ &= A\vec{x}_k - \vec{b} + \alpha_k A\vec{d}_k \\ &= \vec{g}_k + \alpha_k A\vec{d}_k.\end{aligned}$$

Nyt kertomalla yo. yhtälöä termillä $\vec{g}_i^T, i \in \{0, \dots, N-1\}$ vasemmalta saamme

$$\begin{aligned}\vec{g}_i^T \vec{g}_{k+1} &= \vec{g}_i^T \vec{g}_k + \alpha_k \vec{g}_i^T A\vec{d}_k \\ \alpha_k \vec{g}_i^T A\vec{d}_k &= \vec{g}_i^T \vec{g}_{k+1} - \vec{g}_i^T \vec{g}_k \\ \vec{g}_i^T A\vec{d}_k &= \frac{\vec{g}_i^T \vec{g}_{k+1} - \vec{g}_i^T \vec{g}_k}{\alpha_k}.\end{aligned}$$

Vasemmalla yo. yhtälössä on termit, jotka meidän täytyy laskea kertoimia β_{ij} varten. Oikea puoli taas riippuu gradienttien sisätuloista. Valitsemalla α_i :t kuten lauseessa 3.25 tehdään, iteraation pisteet osoittautuvat sellaisiksi, joissa gradientit ovat ortogonaalisia kaikkiin aikaisempiin gradientteihin nähden. Osoitetaan ensin, että gradientit ovat ortogonaalisia kaikkiin aikaisempiin etsintäsuuntiin \vec{d}_i nähden:

$$\begin{aligned}\vec{d}_i^T \vec{g}_j &= \vec{d}_i^T (A\vec{x}_j - \vec{b}) \\ &= \vec{d}_i^T (A\vec{x}_j - A\vec{x}^*) \\ &= \vec{d}_i^T \left(A\vec{x}_0 + \sum_{k=0}^{j-1} \alpha_k A\vec{d}_k - A\vec{x}_0 - \sum_{k=0}^{N-1} \alpha_k A\vec{d}_k \right) \\ &= \vec{d}_i^T \left(\sum_{k=j}^{N-1} \alpha_k A\vec{d}_k \right) \\ &= \sum_{k=j}^{N-1} \alpha_k \vec{d}_i^T A\vec{d}_k \\ &= 0, \quad \text{kun } i < j.\end{aligned}$$

Tästä johtuen, kun $i < j$ pätee lisäksi

$$\begin{aligned}0 &= \vec{d}_i^T \vec{g}_j = \vec{u}_i^T \vec{g}_j + \sum_{k=0}^{i-1} \beta_{ik} \vec{d}_k^T \vec{g}_j = \vec{u}_i^T \vec{g}_j \\ \implies \vec{u}_i^T \vec{g}_j &= 0.\end{aligned}$$

Nyt, koska valitsimme vektorit $\vec{u}_i = \vec{g}_i$, niin pätee $\vec{g}_i^T \vec{g}_j = 0$, kun $i < j$. Siis gradientit ovat ortogonaalisia kaikkien edellisten gradienttien kanssa. Tällöin ne muodostavat lineaarisesti riippumattoman joukon, josta voimme konjugoida etsintäsuunnat \vec{d}_i .

Palataan nyt takaisin Gram-Schmidtin kertoimiin β_{ij} . Edellisten tulosten perusteella pätee

$$\beta_{ik} = \begin{cases} \frac{1}{-\alpha_{i-1}} \frac{\vec{g}_i^T \vec{g}_i}{\vec{d}_{i-1}^T A \vec{d}_{i-1}}, & \text{kun } k = i - 1 \\ 0 & \text{muulloin} \end{cases}$$

Enää emme siis tarvitse kaikkia etsintäsuuntia \vec{d}_i uuden etsintäsuunnan konjugointiin, sillä kyseinen vektori on jo valmiiksi konjugoitu lähes kaikkien edellisten etsintäsuuntien mukaan. Voimmekin itseasiassa määritellä

$$\beta_i := -\frac{1}{\alpha_{i-1}} \frac{\vec{g}_i^T \vec{g}_i}{\vec{d}_{i-1}^T A \vec{d}_{i-1}} = \frac{\vec{d}_{i-1}^T A \vec{d}_{i-1}}{\vec{g}_{i-1}^T \vec{d}_{i-1}} \frac{\vec{g}_i^T \vec{g}_i}{\vec{d}_{i-1}^T A \vec{d}_{i-1}} = \frac{\vec{g}_i^T \vec{g}_i}{\vec{g}_{i-1}^T \vec{d}_{i-1}} = \frac{\vec{g}_i^T \vec{g}_i}{\vec{g}_{i-1}^T \vec{g}_{i-1}}$$

Nyt uuden A -konjugoidun etsintäsuunnan löytäminen yksinkertaistuu kaavan 3.29 mukaan muotoon

$$\vec{d}_i = \vec{g}_i + \beta_i \vec{d}_{i-1}.$$

Kokonaisuudessaan algoritmi on siis seuraavanlainen:

Lause 3.31. *Conjugate gradient theorem:* Olkoon A symmetrinen, positiivisesti definitti $N \times N$ matriisi. Olkoon $\vec{x}_0 \in \mathbb{R}^N$ alkuarvaus. Tällöin kaikilla \vec{x}_0 iteraatio

$$\begin{aligned} \vec{d}_0 &= \vec{g}_0 = A\vec{x}_0 - \vec{b}, \\ \vec{x}_{i+1} &= \vec{x}_i + \alpha_i \vec{d}_i, \\ \alpha_i &= -\frac{\vec{g}_i^T \vec{d}_i}{\vec{d}_i^T A \vec{d}_i}, \\ \vec{d}_{i+1} &= \vec{g}_{i+1} + \beta_{i+1} \vec{d}_i, \\ \beta_{i+1} &= -\frac{\vec{g}_{i+1}^T \vec{g}_{i+1}}{\vec{g}_i^T \vec{g}_i}, \\ \vec{g}_i &= A\vec{x}_i - \vec{b}, \end{aligned}$$

suppenee lineaarisen systemin $A\vec{x} = \vec{b}$ ratkaisuun \vec{x}^* N :llä iteraatiolla, siis $\vec{x}_N = \vec{x}^*$.

3.10 CGLS:n soveltaminen normaaliyhtälöihin

Ongelmaksi jää vielä se, että lineaarisen systeemin 3.10 matriisi A ei käytännössä koskaan ole neliömatriisi. Tähän ratkaisuksi hyödynnämme algoritmia ns. *normaaliyhtälöihin*

$$(3.32) \quad A\vec{x} = \vec{b} \iff A^T A\vec{x} = A^T \vec{b}$$

Matriisi $A^T A$ on selvästi symmetrinen ja ainakin positiivisesti semidefiniitti, sillä kaikilla $\vec{z} \neq \vec{0}$ pätee

$$\vec{z}^T A^T A \vec{z} = (A\vec{z})^T A\vec{z} = \|A\vec{z}\|^2 \geq 0.$$

Haluaisimme kuitenkin matriisin $A^T A$ olevan positiivisesti definiitti, eli $\|A\vec{z}\|^2 > 0$ kaikilla $\vec{z} \neq \vec{0}$. Tutkitaan vielä lyhyesti tilanteita, jolloin $\|A\vec{z}\|^2 = 0 \iff \vec{z}^T A^T A \vec{z} = 0$.

Olkoon matriisin A sarakkeet lineaarisesti riippuvia. Tällainen tilanne saattaa syntyä esimerkiksi, kun yksi röntgensäde lävistää eri vokselit samoilla pituuksilla. Myös, kun jokin vokselia ei lävistä yksikään röntgensäde, matriisiin A syntyy nollasarake, jolloin sen sarakkeet ovat lineaarisesti riippuvat. Merkitään vektorilla \vec{a}_i :llä matriisin A i :ttä saraketta, siis $A = [\vec{a}_1 \ \dots \ \vec{a}_N]$. Koska vektorit \vec{a}_i ovat lineaarisesti riippuvia, on olemassa kertoimet $c_i \in \mathbb{R}$, joista ainakin osa ei ole nollia, joille pätee

$$c_1 \vec{a}_1 + \dots + c_N \vec{a}_N = \vec{0}.$$

Merkitään nyt vektorilla $\vec{c} = [c_1 \ \dots \ c_N]^T$ kertoimia c_i . Tällöin

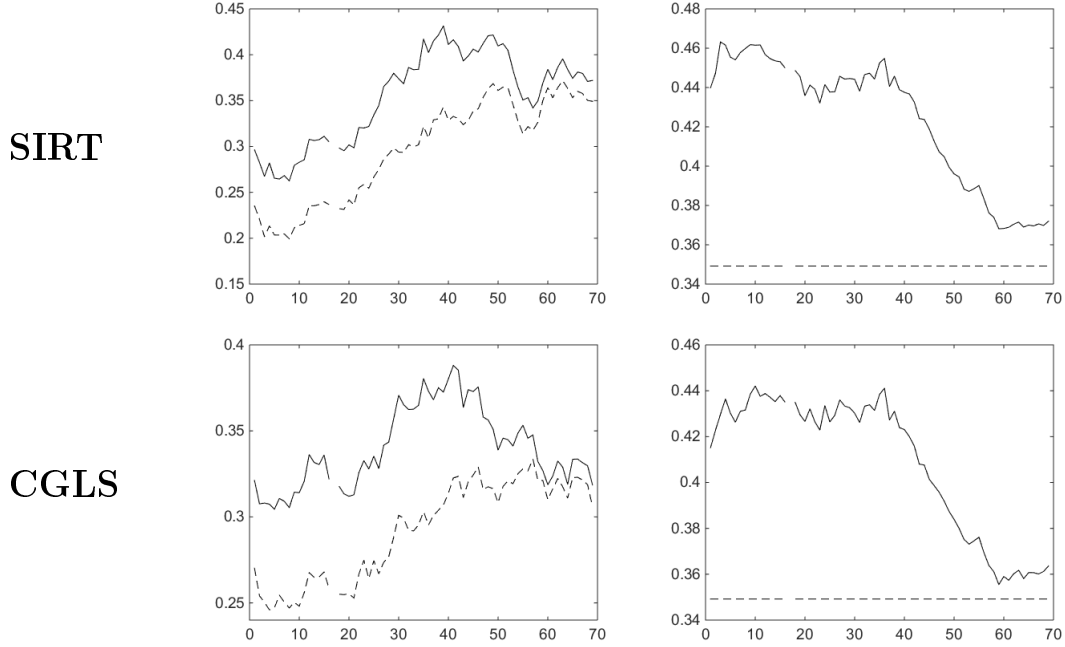
$$A\vec{c} = \begin{pmatrix} c_1 a_{11} + \dots + c_N a_{1N} \\ \vdots \\ c_1 a_{1N} + \dots + c_N a_{NN} \end{pmatrix} = c_1 \vec{a}_1 + \dots + c_N \vec{a}_N = \vec{0}.$$

Olemme siis löytäneet vektorin $\vec{c} \neq \vec{0}$, jolle pätee

$$\vec{c}^T A^T A \vec{c} = \|A\vec{c}\|^2 = 0.$$

Tällöin matriisi A ei ole positiivisesti definiitti. Emme voi siis taata, että matriisi A täyttää kaikki algoritmin oletukset. Ongelmaa helpottaa toisaalta se, että tunnemme mittausedometrian ja esimerkiksi matriisin A nollarivit ja -sarakkeet on mahdollista poistaa ennen rekonstruktioita. Teoriassa matriisi A ei siis usein täytä algoritmin vaatimuksia. Käytännössä kuitenkin CGLS on osoittanut toimivuutensa röntgenrekonstruktioita laskettaessa.

CGLS-algoritmin tulisi saavuttaa tarkka ratkaisu vain N :llä iteraatiolla, mutta mittausrvirheistä ja laskennan epätarkkuudesta johtuen algoritmi ei käytännössä koskaan tee niin. Kuitenkin jo suhteellisen pienillä konjugoitujen vektorien määrällä voidaan saavuttaa melko tarkka approksimaatio systeemin ratkaisulle.



Kuva 3.6: Vasemmalla eri rekonstruktioiden kohinatasot ϵ_t kohdassa, jossa ei pitäisi olla materiaalia (katkoviiva) ja vastaava keskiarvo muovisessa kohdassa (yhtenäinen viiva). Kohdassa $t = 17$ on epäjatkuvuuskohta, sillä röntgendataa tälle ajanhetkelle ei voitu käyttää. Oikealla vastaavat arvot korjauksen jälkeen.

3.11 Aikaulottuvuuden normalisointi

Edellä on esitetty kaksi eri algoritmia, joilla tässä opinnäytetyössä on laskettu röntgenrekonstruktio. Rekonstruktioiden vokselien attenuaatioarvot kuitenkin heittelevät melko paljon ajan mukana. Rekonstruktioalgoritmit tuottavat samoille vokseleille erilaisia attenuaatioarvoja eri ajanhetkille, sillä algoritmeissa ei oteta ollenkaan huomioon aikaulottuvuutta. Lisäksi eri rekonstruktioalgoritmit tuottavat samalle materiaalille eri attenuaatioarvoja. Myös mittauksissa esiintyvä kohina tuottaa hajontaa attenuaatioarvoihin.

Tutkimalla kuvaajia 3.6, näemme, että SIRT-rekonstruktioiden attenuaatioarvot ovat hieman korkeampia kuin CGLS-rekonstruktioiden. Lisäksi viimeisen rekonstruktion kohinavokselit ovat attenuaatioarvoltaan suurempia kuin ensimmäisen rekonstruktion muoviset vokselit. Tämän takia ensimmäisen ajanhetken rekonstruktio hukkuisi animaatiassa kohinaan suhteessa viimeiseen ajanhetkeen ilman normalisointia. Jokaisella ajanhetkellä kuitenkin on löydettävissä selkeä ero kohinavokselien ja materiaalia sisältävien vokselien välillä, joten ongelma on ratkaistavissa skaalaamalla kohinatasot yhtenäisiksi.

Ajanhetken t kohinataso $\epsilon_t \in [0, 1]$ on arvo, joka kuvaa keskimääräistä kohinavokse-

lin attenuaatioarvoa. Kohinataso on 69. SIRT-rekonstruktion 268. kerroksen vasemman yläkulman vokselien, rivien ja sarakkeiden 1-20 keskiarvo. Tämä arvo on käsin valittu, sillä tässä kohdassa ei missään rekonstruktiossa ole materiaaliosia. Kyseisessä kohdassa ei myöskään ole huomattavia rekonstruktioartefakteja. Tällä metodilla saamme yleiseksi kohinatasoksi arvon $\epsilon_0 = 0,349196078431372$.

Skaalaminen toteutetaan siten, että otamme yleisen kohinatason ϵ_0 vertailuarvoksi ja skaalaamme kaikkien vokselien attenuaatioarvoja ylös- tai alaspäin, jotta kohinatasot kohtaavat:

$$f_{i,t} = \frac{\epsilon_t}{\epsilon_0} f_{i,t}.$$

Tutkimalla kohinavokseleiden ja materiaalivokseleiden eroa skaalaamisen jälkeen, kuvaa-
jasta 3.6 nähdään, että kaikki vokselit, joiden attenuaatioarvo on pienempi kuin ϵ_0 voidaan luokitella kohinaksi ja loput eriasteisiksi materiaaliosiksi.

Luku 4

Tulokset ja pohdintaa

Soveltamalla esiteltyjä ohjelmistoja tutkielman aineistoon, pystymme luomaan 4D-rekonstruktion LEGO-robotista. Rekonstruoidun vokseliverkon tiheys on $512 \times 512 \times 512$ ja käytettyjen röntgenkuvien resoluutio on 560×592 . Lopulliset animaatiot ovat esillä osoitteessa YouTube-kanavallani osoitteessa <https://www.youtube.com/channel/UCN4eD4a4nAMUMJOINOTQ3-Q>.

Yksittäisten ajanhetkien rekonstruktioita tutkimalla nähdään, että molemmat algoritmit toimivat toivotulla tavalla. Rekonstruktiot tavoittavat robotin yleiset piirteet todella hyvin. Molemmat rekonstruktiot ovat myös todella tarkkoja. Pienetkin yksityiskohdat erottuvat selvästi. Esimerkiksi kuvassa 4.2 oleva syvennys on vain noin millimetrin syvyinen. Saman osan pääty on ontto ja seinämät todella ohuet. Kuvasta 4.3 näkyy hyvin, että rekonstruktiot tavoittavat myös nämä robotin ohuemmat osat. Myös itse animaatiot näyttävät siltä, miltä pitääkin. LEGO-osat pyörivät akseleidensa ympärillä ja robotti pysyy rekonstruktioissa lähes paikallaan.

CGLS-algoritmi näyttäisi saavuttavan paremman kontrastin kuin SIRT. Esimerkiksi kuvista 4.1 ja 4.5 näkyy, että CGLS-rekonstruktio on selkeästi terävämpi.

Tutkimalla kuvissa 4.1 näkyviä poikkileikkauksia, pystytään jopa erottamaan eri LEGO-osat, vaikka ne ovat tiiviisti kiinni toisissaan. Keskimmäinen, vaaleampi alue on pystysuuntainen akseli. Akselissa on kiinnitetty LEGO-palikka, josta lähtee vastaavat akselit kolmeen eri suuntaan, 120 asteen välein. Näiden päihin taas on kiinnitetty erilaiset osat.

Poikkileikkauksia tutkimalla nähdään myös, että tutkittava robottimme sisältää kahdentyypistä muovia, sillä kuvista voidaan erottaa kaksi eri attenuaatiotasoa. Suurin osa käytetyistä LEGO-osista on yhtä materiaalia, mutta kuvissa 4.1 nähdään molemmissa keskellä ristin muotoinen, selvästi ympäröivää aluetta vaaleampi alue. Myös haarakkeissa nähdään muuta kappaletta kirkkaampi viiva. Tällöin näillä alueilla on oltava korkeampi attenuaatiokerroin kuin muilla LEGO-osilla.

Korkeamman attenuaatiotason omaavat kappaleet ovat akseleita, jotka yhdistävät robotin eri osia toisiinsa. Lisäksi akselit vaikuttaisivat olevan onttoja. Poikkileikkauksista

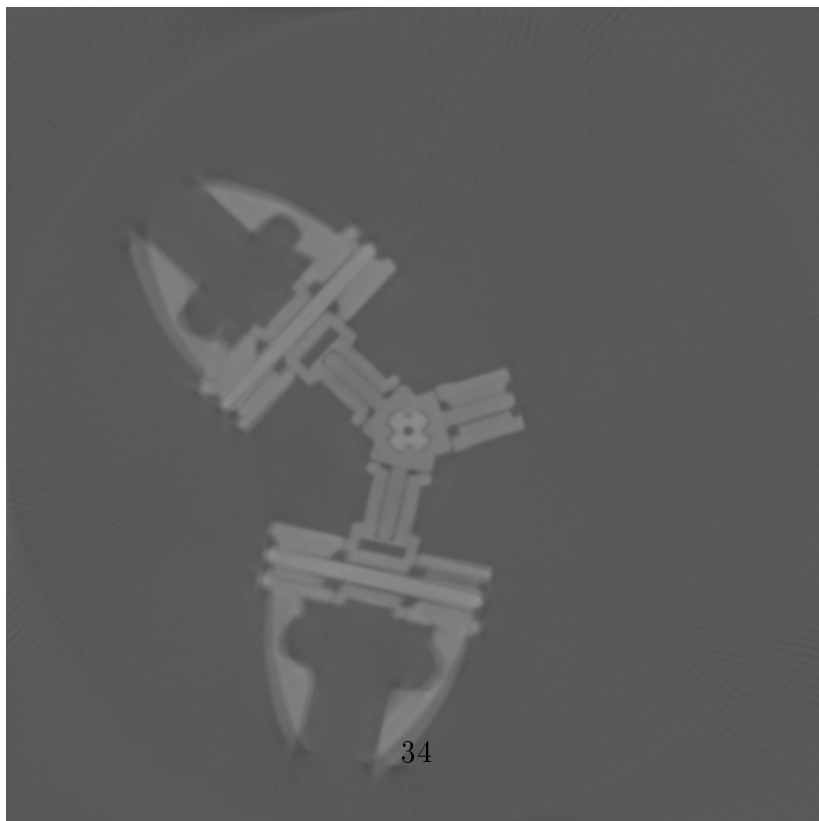
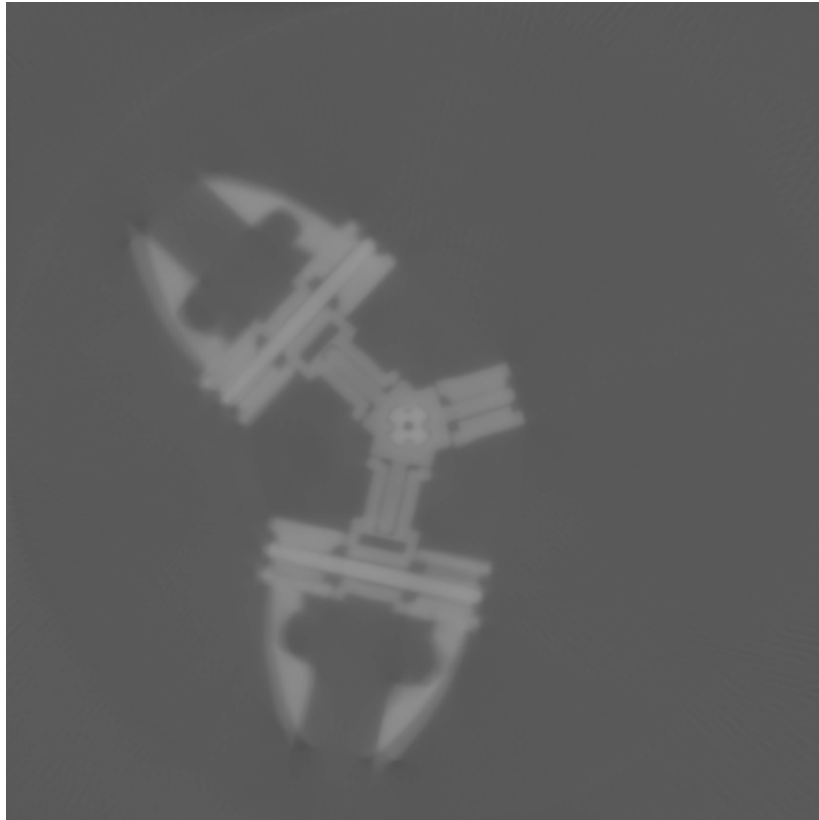
nähdään, että pystysuuntainen akseli on keskeltä tyhjää täynnä. Muita poikkileikkauksia tutkimalla nähdään, että myös vaakasuorassa olevat akselit ovat keskeltä onttoja. Akselit on visualisoitu kuvissa 4.6.

Osat, jotka sijaitsevat lähellä kuution keskipistettä ovat rekonstruktioissa tarkempia kuin osat, jotka ovat kuution reunoilla. Tutkimalla ajanhetkeä 18 kuvasta 4.4 nähdään, että pienempi pyörivä osa vääristyy selvästi sen käydessä lähellä kuution reunaa. Lähellä pohjaa olevat osat ovat myös selvästi vääristyneitä. Näille havainnoille yhteinen selitys on mittauserometriassa aiheuttamat vääristymät. Mitä kauempana vokseli on rekonstruoitavan alueen keskipisteestä, sitä harvempi röntgensäde kulkee sen lävitse. Näille vokseleille on siis vähemmän dataa käytettävissä.

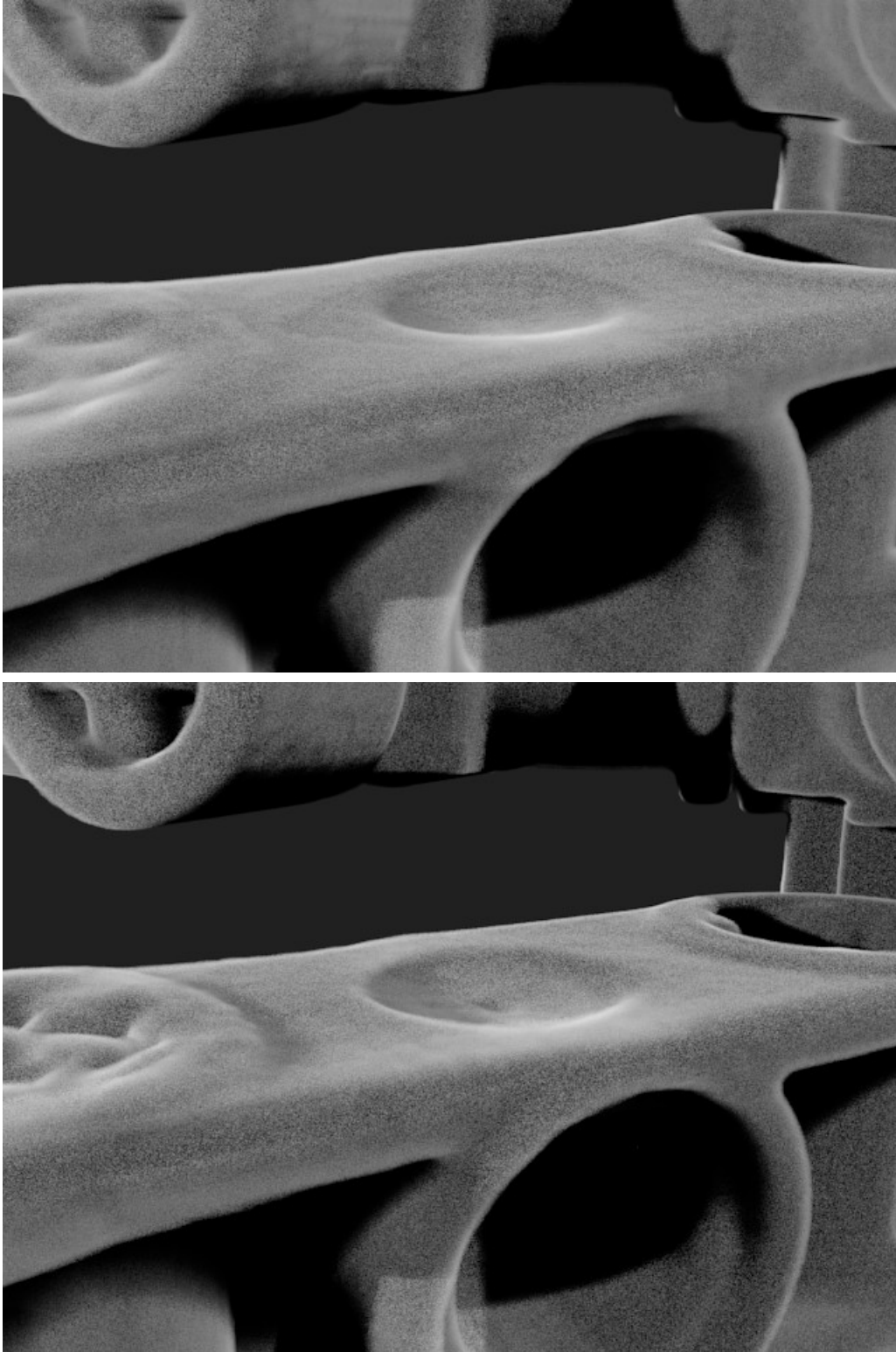
Lisäksi mittauserometriassa saattaa olla pieniä kalibrointivirheitä, esimerkiksi röntgenkuvien keskiviiva ei välttämättä leikkaa tarkasti robotin pyörimiskeskipistettä. Tällöin röntgenkuvat eivät ole tarkasti asemoitu, jolloin uloimpien vokseleiden tarkkuus kärsii. Mittalaitteiston kalibrointivirheet aiheuttavat myös muotojen kahdentumista. Esimerkiksi kuvassa 4.1, haarakkeiden kärjet selvästi kahdentuvat. Kahdentuminen pahenee, mitä kauemmaksi pyörimiskeskipisteestä mennään.

Kuution alareunassa nähdään myös selvä attenuaatiokeskittymä yhdellä reunalla. Tämä johtuu siitä, että tällä korkeudella sijaitsi metallisia osia, jotka eivät kuulu rekonstruoitavaan alueeseen. Joissakin röntgenprojektioissa ne kuitenkin näkyvät, joten alimpien poikkileikkauksien vokseleihin syntyy tästä syystä lisää attenuaatiota.

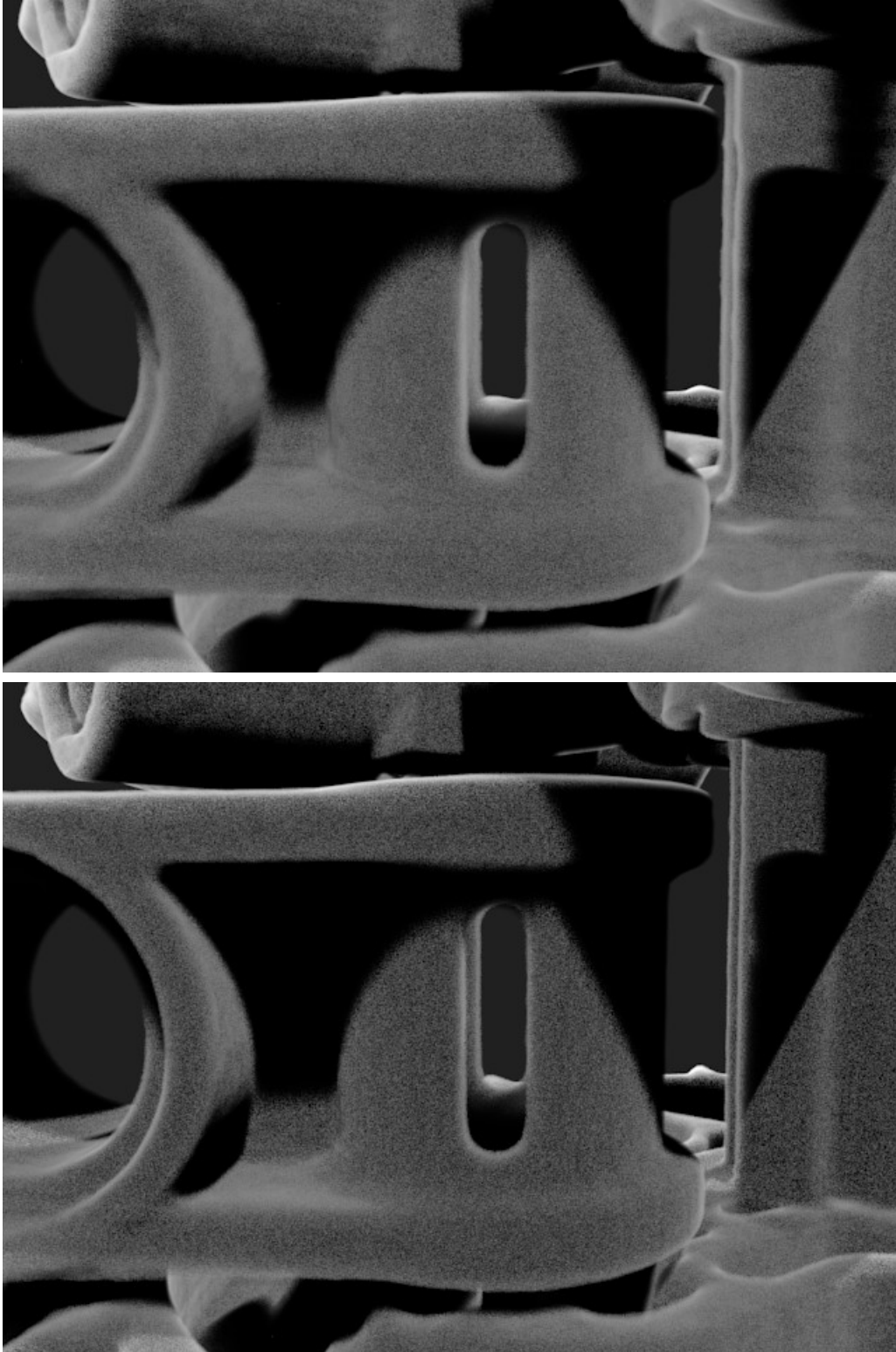
Lisäksi rekonstruktioissa on nähtävissä ympyräartefakteja. Kuvassa 4.1 tällainen on nähtävissä vasemmalla ylhäällä melko haaleana. Näille voi myös olla monia syitä. Yksi syy tähän on se, että rekonstruoitavan kuution reuna-alueille ei ole saatavilla enää täyttä röntgendataa. Kappaleen pyöriessä nämä alueet näkyvät vain joissakin projektioissa, joten niiden muodostama attenuaatio jää hieman vajaaksi.

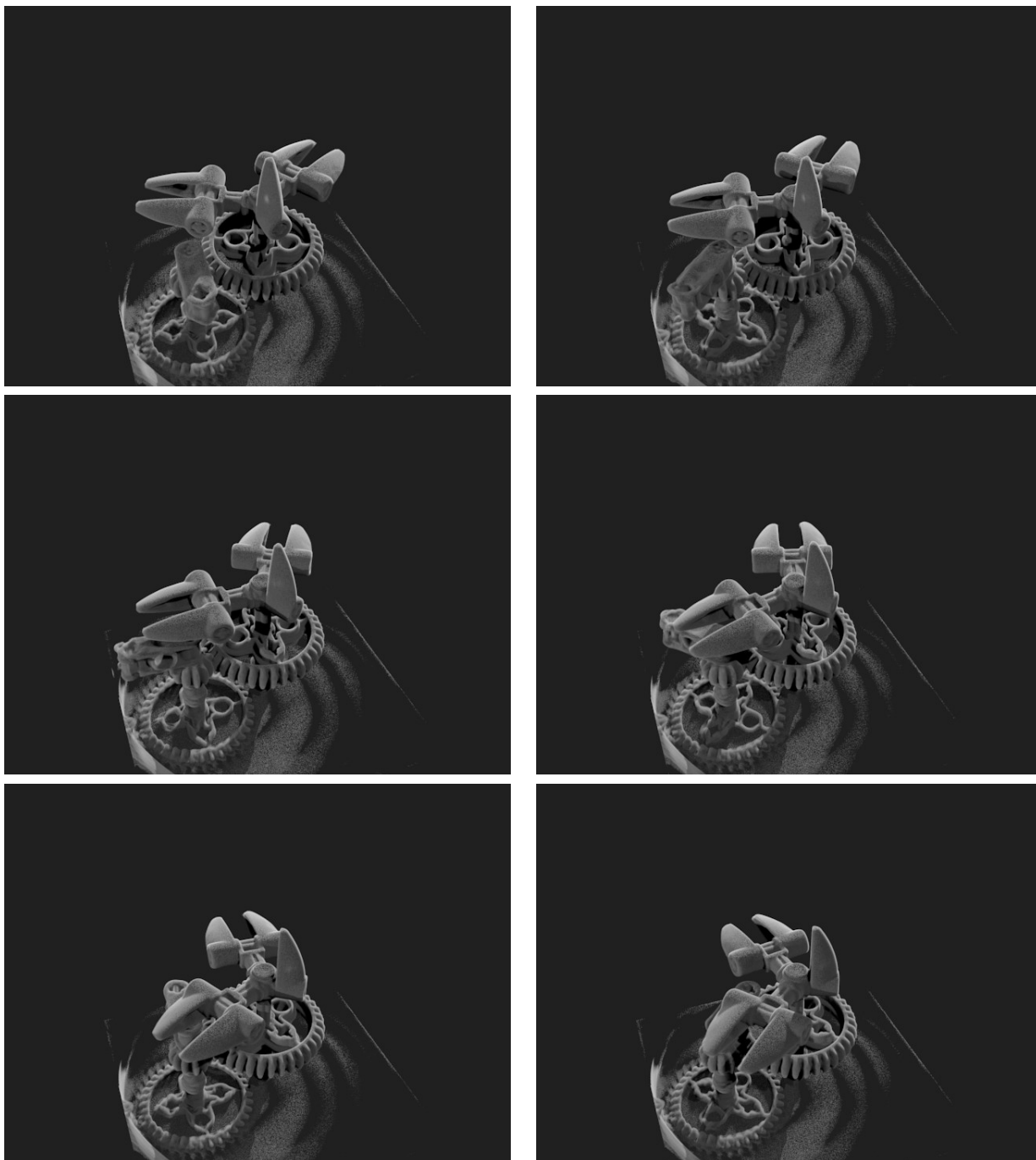


Kuva 4.1: Ensimmäisen ajanhetken 268. poikkileikkaus. Ylempänä SIRT ja alempana CGLS.

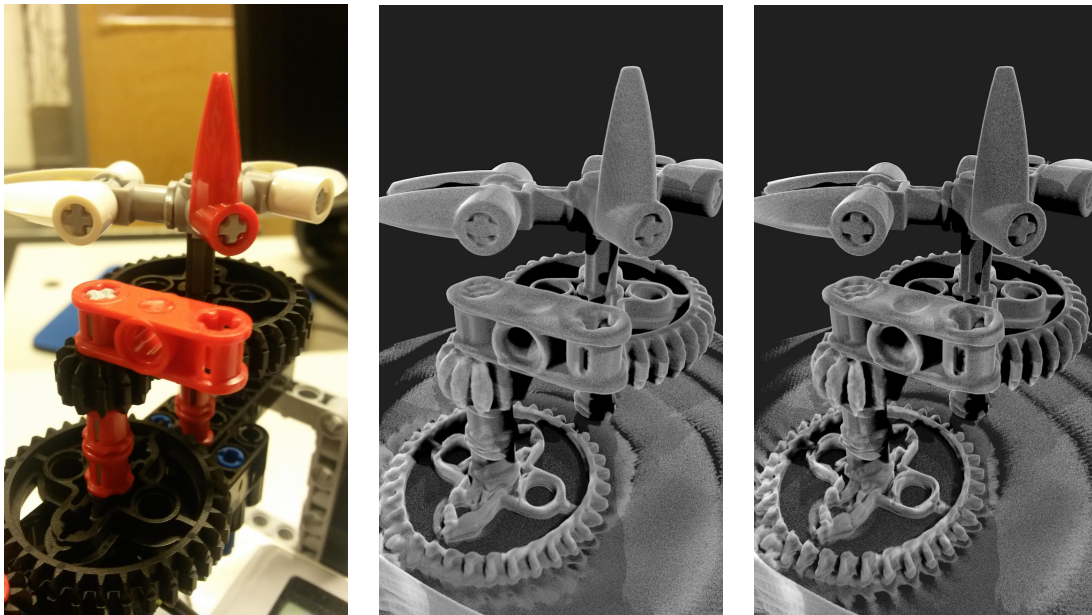


35
Kuva 4.2: Noin millimetrin syvennys yhdessä LEGO-osassa. Ylempänä SIRT ja alempana CGLS.

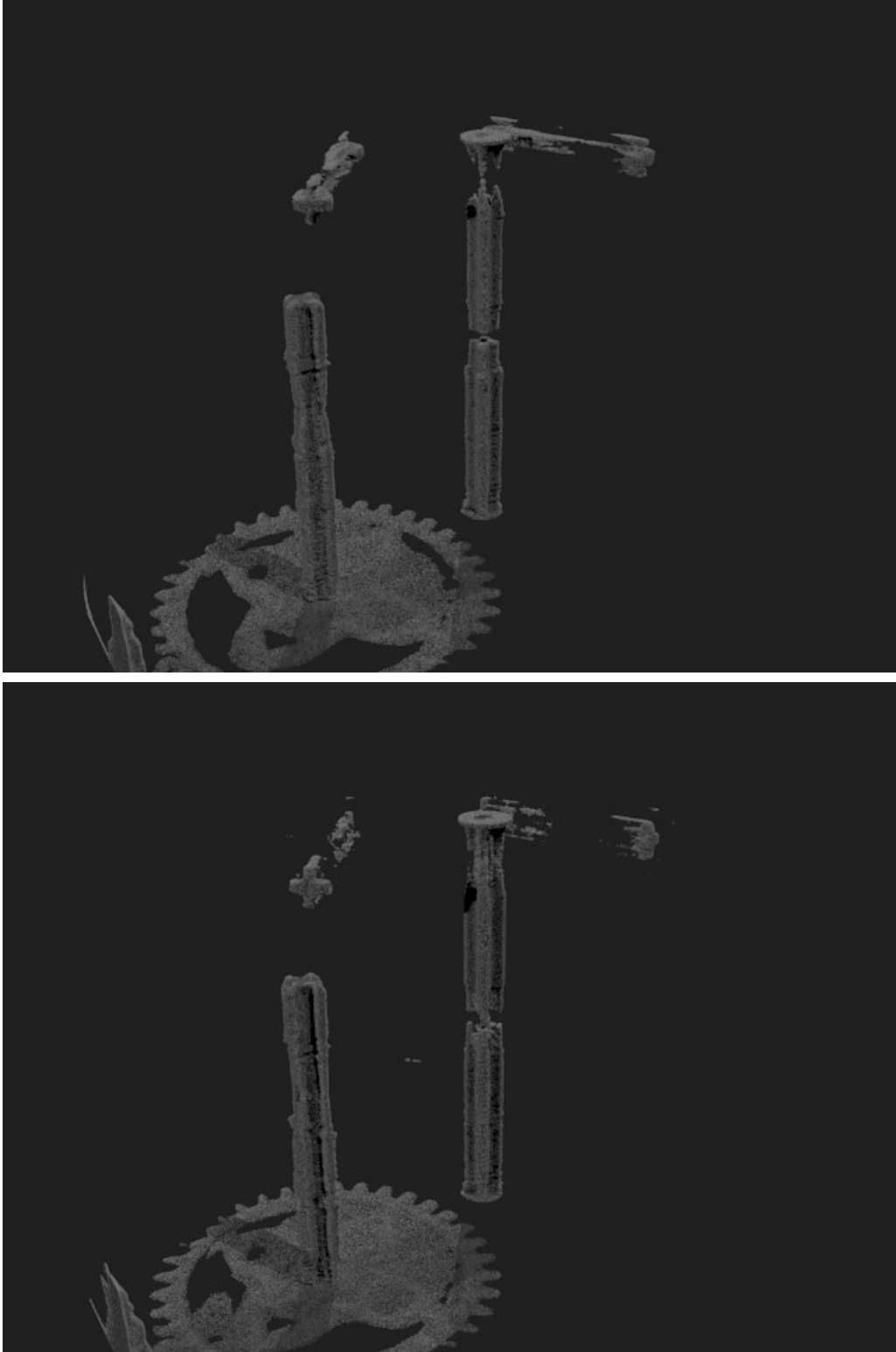




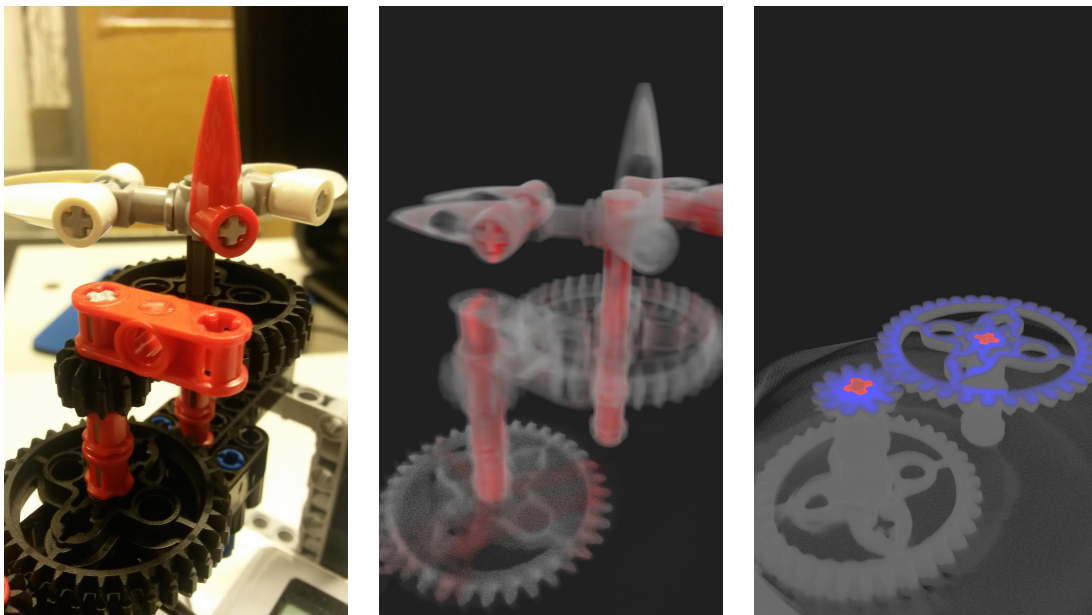
Kuva 4.4: Ajanhetket 10, 14, 18, 22, 26 ja 30 lopullisesta animaatiosta, SIRT-rekonstruktio.



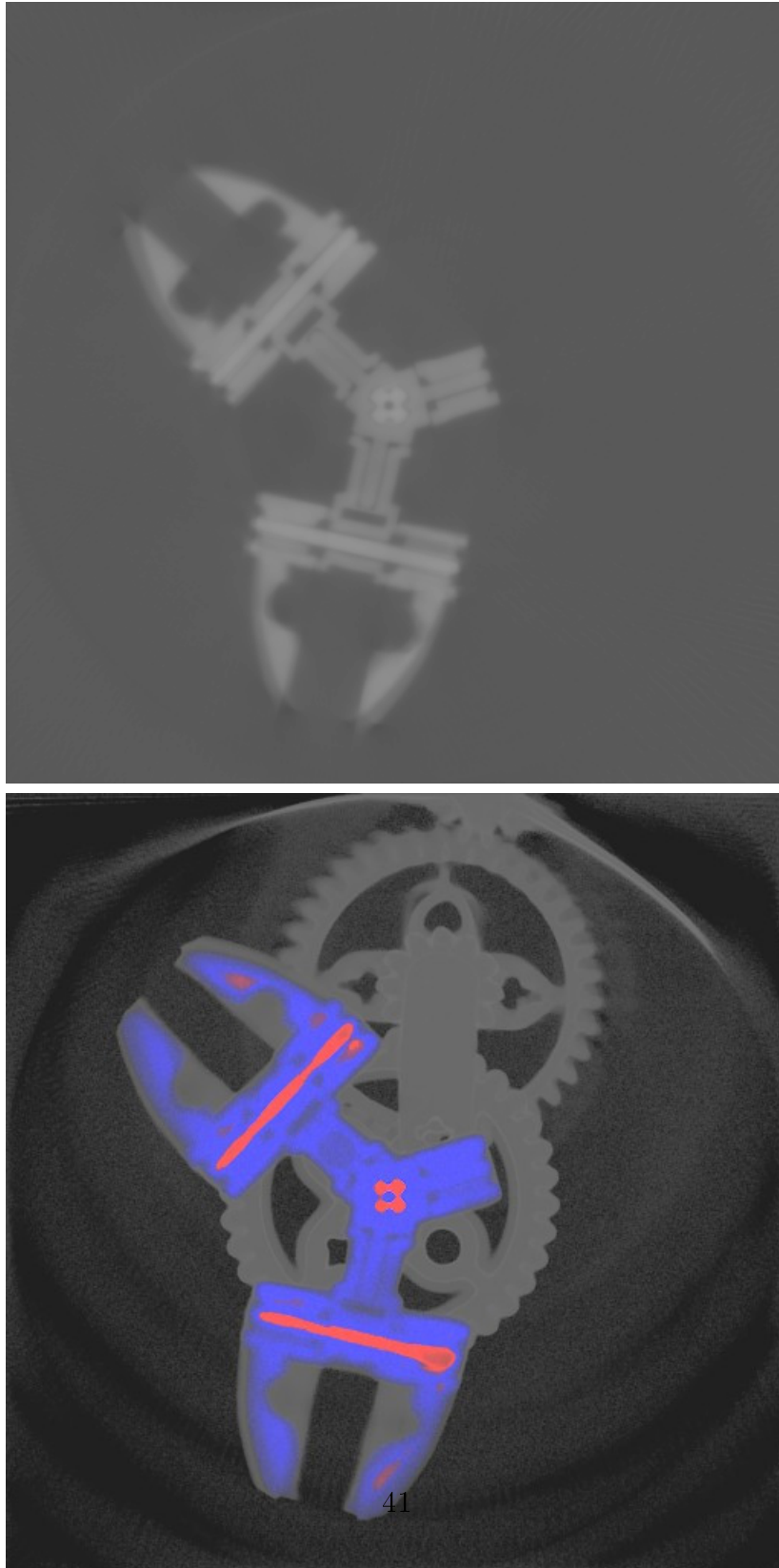
Kuva 4.5: Rekonstruktioiden vertailu aitoon kuvaan. Kuva on otettu mittauksen jälkeen, joten robotti ei kuvassa ja rekonstruktioissa ole aidosti samassa asennossa. Ajanhetkellä 7 robotti on kuitenkin suhteellisen samassa asennossa, joten rekonstruktioita voidaan verrata kuvaan. SIRT-algoritmilla lasketulla rekonstruktioilla (kesk.) näyttäisi olevan enemmän kohinaa kuin CGLS:llä (oik.), jonka kontrasti on hieman parempi. Yleisesti ottaen molemmat algoritmit kykenevät erottelamaan pienetkin yksityiskohdat robotista.



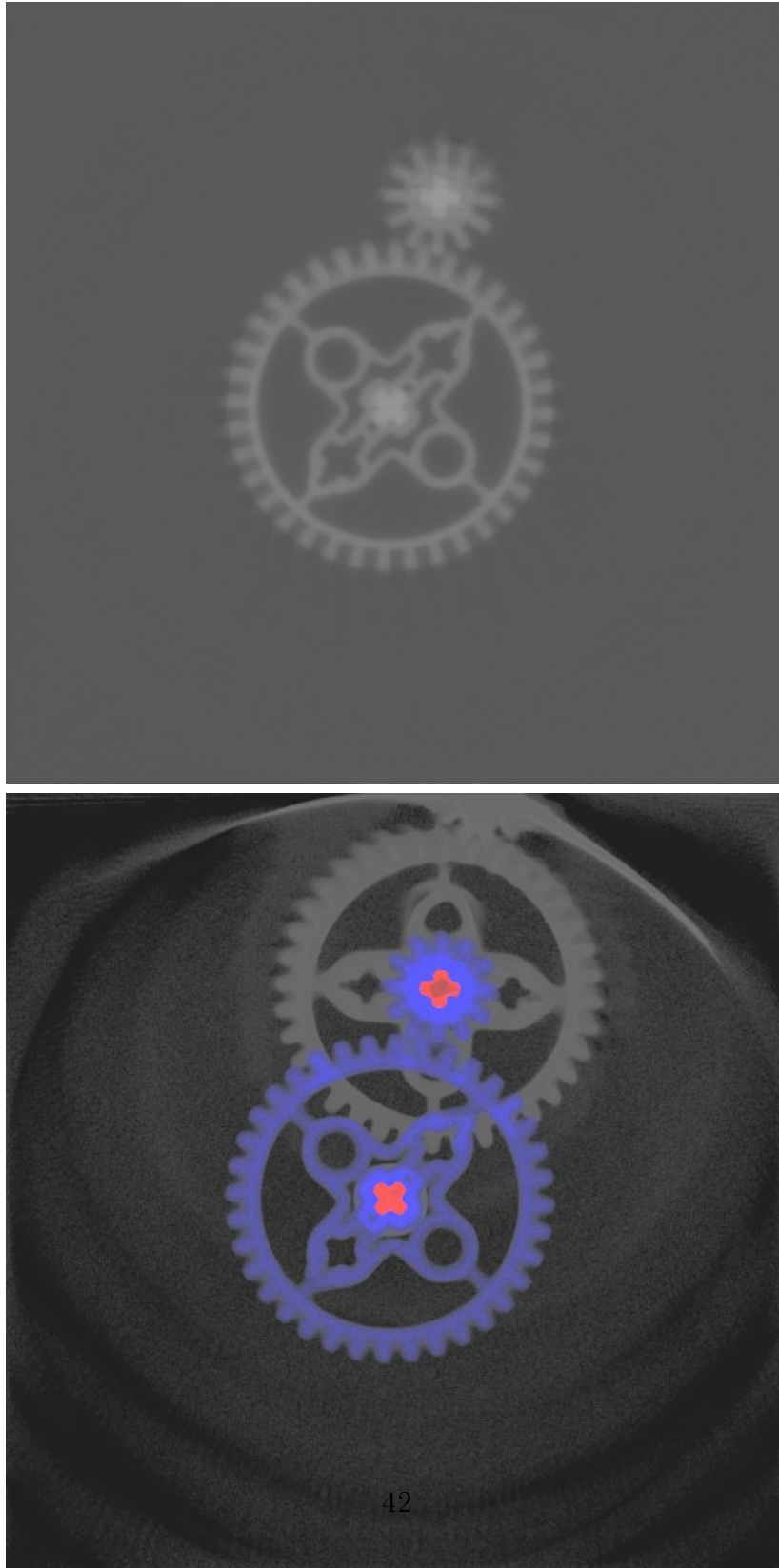
39
 Kuva 4.6: Robotin kaksi erillistä pysty- ja kaksi vaaka-akselia voidaan erottaa piirroksissa omiksi kappaleikseen korkeamman attenuaatiotasonsa vuoksi. Akseleiden attenuaatiotaso ei säily yhtenäisenä rekonstruktioissa. Kerroksissa, joissa on myös paljon muutakin attenuaatiota, myös akseleiden attenuaatiotaso on alhaisempi. Ylempänä SIRT ja alempana CGLS.



Kuva 4.7: Akselit visualisoituna kuvaan verrattuna. Rekonstruktioissa pystyy hyvin erottamaan kaksi attenuaatiotasoa: tiheämmän akseleiden muovin, joka on visualisoitu punaisena ja harvemman muun muovin, joka on visualisoitu sinisenä.



Kuva 4.8: Ylempänä SIRT rekonstruktion 268. kerros ja alempana sama kerros visualisoi-
tuna 3D-mallissa.



Kuva 4.9: Ylempänä SIRT rekonstruktion 390. kerros ja alempana sama kerros visualisoituna 3D-mallissa.

Kirjallisuutta

- [1] *Linear and Nonlinear Inverse Problems with Practical Applications*, Jennifer L. Mueller, Samuli Siltanen, SIAM, 2012
- [2] *Computed Tomography: From Photon Statistics to Modern Cone-Beam CT*, Thorsten M. Buzug, Springer, 2008
- [3] *Principles of Computerized Tomographic Imaging*, A. C. Kak, Malcolm Slaney, Society of Industrial and Applied Mathematics, 2001
- [4] *The ASTRA Toolbox: a platform for advanced algorithm development in electron tomography*, W. Van Aarle, W. J. Palenstijn, J. De Beenhouwer, T. Altantzis, S. Bals, K. J. Batenburg, and J. Sijbers, Ultramicroscopy, vol. 157, pp. 35-47, 2015
- [5] *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*, Edition 1 $\frac{1}{4}$, Jonathan Richard Shewchuk, <https://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf>, 1994
- [6] *Dynamic multi-source X-ray tomography using a spacetime level set method*, E. Nieminen, M. Lassas, A. Kallonen, L. Harhanen, K. Hämäläinen, S. Siltanen, Journal of Computational Physics, Volume 291 Issue C, Pages 218-237, June 2015
- [7] *Design and Construction of an X-ray Computed Tomography Imaging System*, A. Meaney, Helsingin yliopisto, Fysiikan laitos, 2015